



UNIVERSITAS SEMARANG
FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI
TEKNIK INFORMATIKA

Mobile Programming

Modul Praktikum Mahasiswa

Oleh:
Alauddin Maulana Hirzan, S. Kom., M. Kom
NIDN. 0607069401

Daftar Isi

Pendahuluan	2
0.1 Mengenal Android	2
0.2 Mengenal Android Studio	2
0.3 Mengenal Android SDK	3
0.4 Mengenal Android Emulator	3
Persiapan Praktikum	5
0.5 Perangkat Keras	5
0.6 Perangkat Lunak	5
1 Praktikum 1	6
1.1 Konfigurasi Android Studio dan Android Emulator	6
1.2 Tutorial	6
2 Praktikum 2	16
2.1 Antarmuka Halaman Login	16
2.2 Tutorial	16
3 Praktikum 3	27
3.1 Pemrograman Halaman Login	27
3.2 Tutorial	27
4 Praktikum 4	35
4.1 Pemrograman Halaman Pendaftaran	35
4.2 Tutorial	35
5 Praktikum 5	43
5.1 SQLite Insert Data	43
5.2 Tutorial	43
6 Praktikum 6	50
6.1 SQLite Read Data	50
6.2 Tutorial	50
7 Praktikum 7	59
7.1 SQLite Update Data	59
7.2 Tutorial	59

8 Praktikum 8	69
8.1 Google Map	69
8.2 Tutorial	69

Daftar Gambar

1	Perangkat Android	2
2	Android Studio	3
3	Android SDK	3
4	Android Emulator	4

Pendahuluan

0.1 Mengenal Android

Sistem operasi Android adalah sistem operasi seluler untuk digunakan terutama untuk perangkat layar sentuh, ponsel, dan tablet. Desainnya memungkinkan pengguna memanipulasi perangkat seluler secara intuitif, dengan gerakan jari yang mencerminkan gerakan umum, seperti mencubit, menggesek, dan mengetuk.



Gambar 1: Perangkat Android

0.2 Mengenal Android Studio

Sebuah *Integrated Development Environment* (IDE) yang digunakan untuk mengembangkan aplikasi Android. IDE ini merupakan IDE yang diperuntukkan untuk pemula karena sederhana dan tidak perlu menambahkan hal-hal kompleks secara manual untuk aplikasi sederhana.



Gambar 2: Android Studio

0.3 Mengenal Android SDK

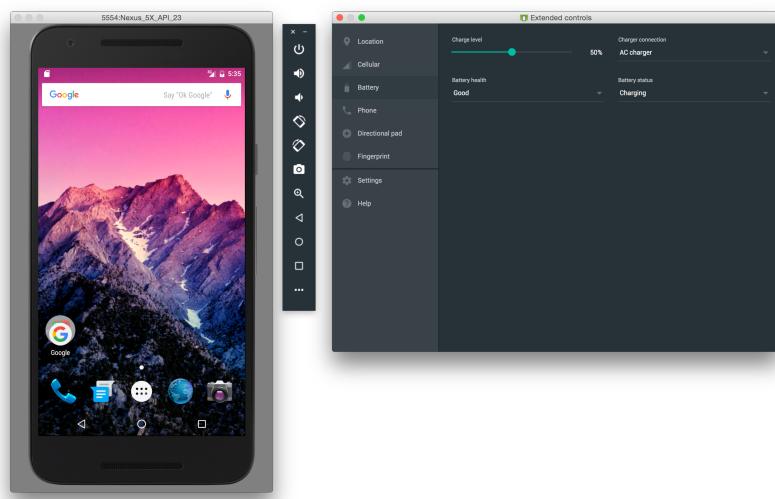
Merupakan framework dasar untuk membuat aplikasi Android dan sudah tersedia secara default di Android Studio. Android SDK memiliki berbagai macam versi tergantung dari versi Android itu Sendiri. Penggunaan Android SDK juga akan mempengaruhi kompatibilitas ke belakang dengan Android versi lama.



Gambar 3: Android SDK

0.4 Mengenal Android Emulator

Android Studio juga menyediakan Android Emulator yang digunakan untuk menghubungkan perangkat Android Virtual ke IDE. Meskipun dianggap cukup berat, namun Emulator ini dapat membantu proses pembuatan aplikasi Android bagi mereka yang tidak memiliki perangkat Android.



Gambar 4: Android Emulator

Persiapan Praktikum

Agar praktikum dapat berjalan dengan lancar, mahasiswa diwajibkan memenuhi persyaratan berikut baik dalam bentuk perangkat keras maupun lunak:

0.5 Perangkat Keras

- Prosesor dengan 4 inti
- RAM minimal 4GB, rekomendasi 8GB
- HDD 10GB

0.6 Perangkat Lunak

Perangkat lunak berikut ini wajib diinstall oleh mahasiswa demi lancarnya praktikum:

- Android Studio terbaru
 - Download via <https://developer.android.com/studio>
- Universal ADB Driver
 - Download via <https://adb.clockworkmod.com/>

Bab 1

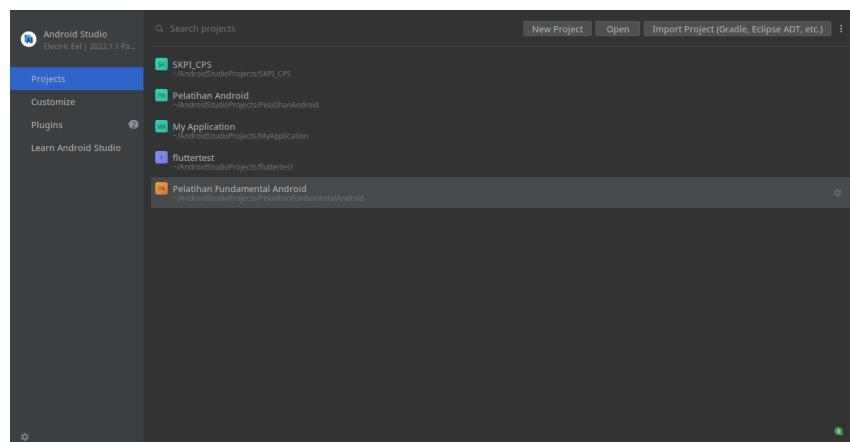
Praktikum 1

1.1 Konfigurasi Android Studio dan Android Emulator

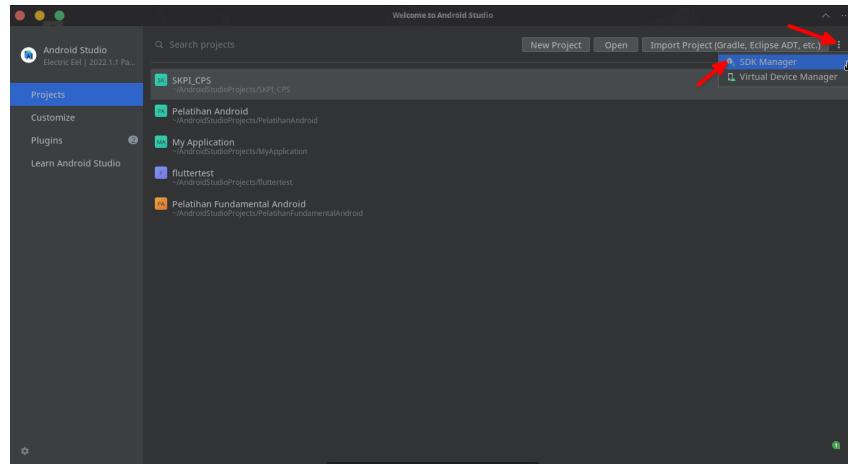
Di bagian ini mahasiswa diajarkan bagaimana mengkonfigurasikan Android Studio khususnya pada bagian Android SDK, dan Android Emulator di komputer masing-masing. Hal ini sangat direkomendasikan agar mahasiswa dapat mengenali struktur Android Studio dengan lebih mudah,

1.2 Tutorial

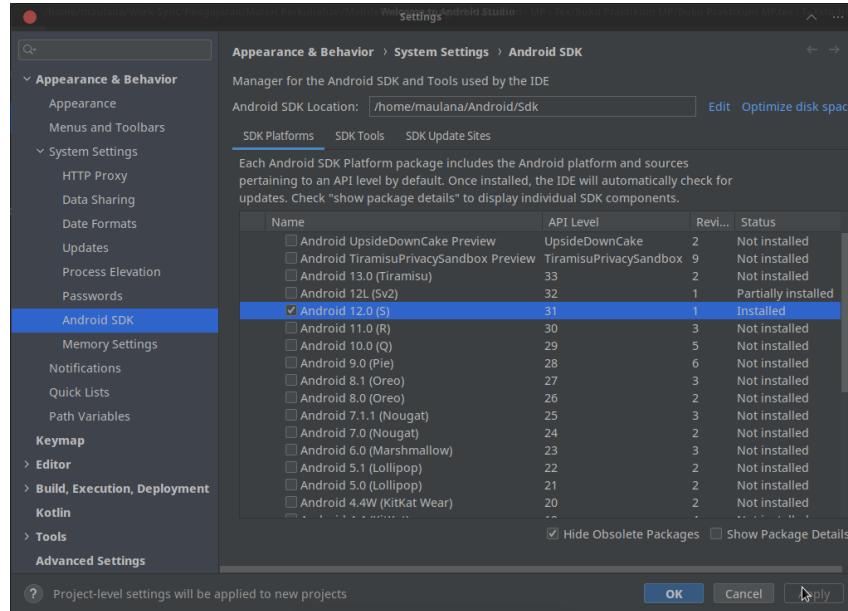
1. Buka **Android Studio**, dan pastikan menggunakan **Versi Terbaru** untuk memulai praktikum ini.



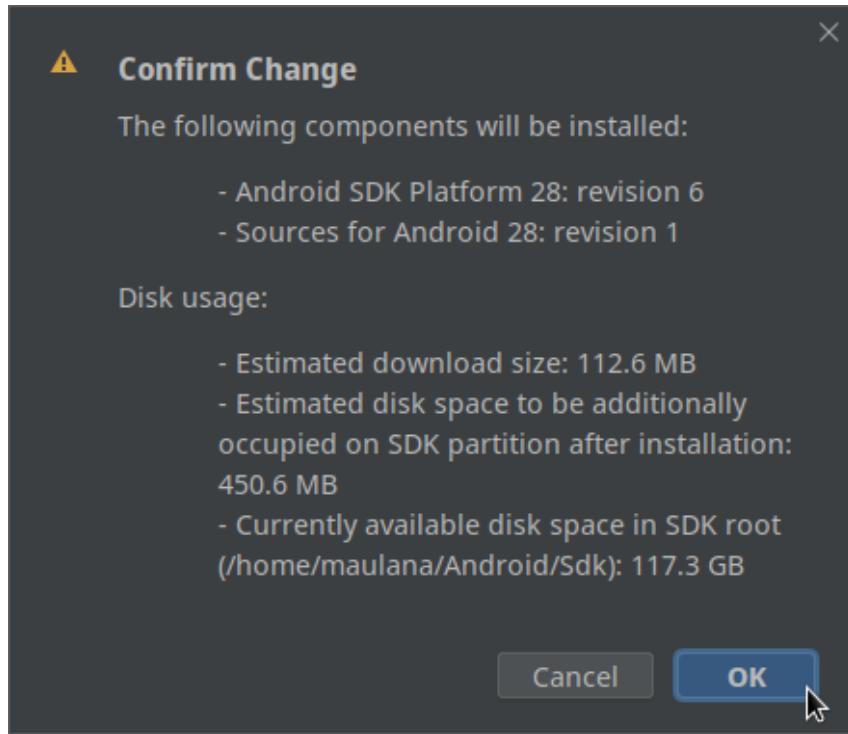
2. Untuk mengkonfigurasikan Android SDK, klik **Titik Tiga / More Action** → **SDK Manager**



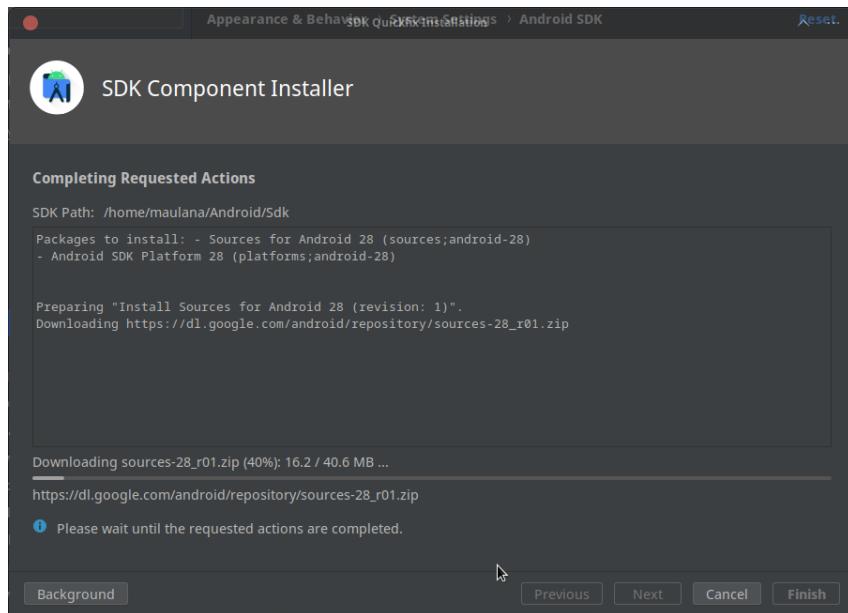
3. Jika sudah muncul tampilan **SDK Manager**, cek apakah ada SDK yang sudah terinstall. Secara **Default**, SDK sudah terpasang secara otomatis bersamaan dengan instalasi Android SDK.



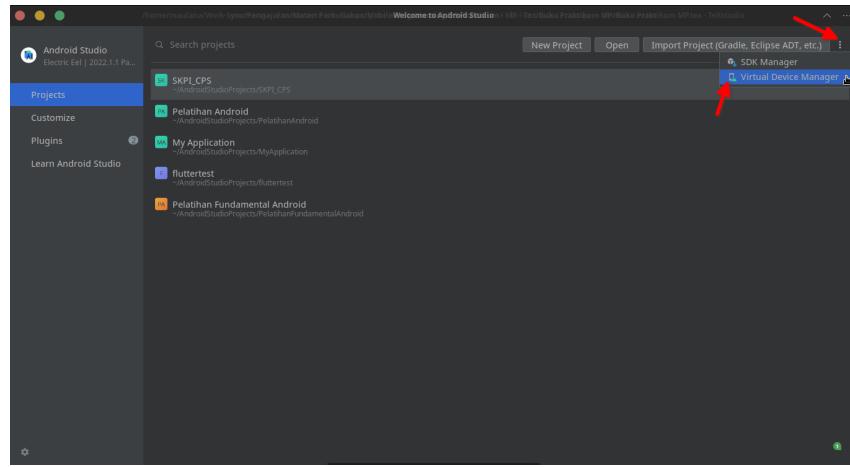
4. Untuk menginstall SDK secara manual, cukup centang SDK target dan klik **Apply**. Android Studio akan memunculkan peringatan (jika ada, klik OK)



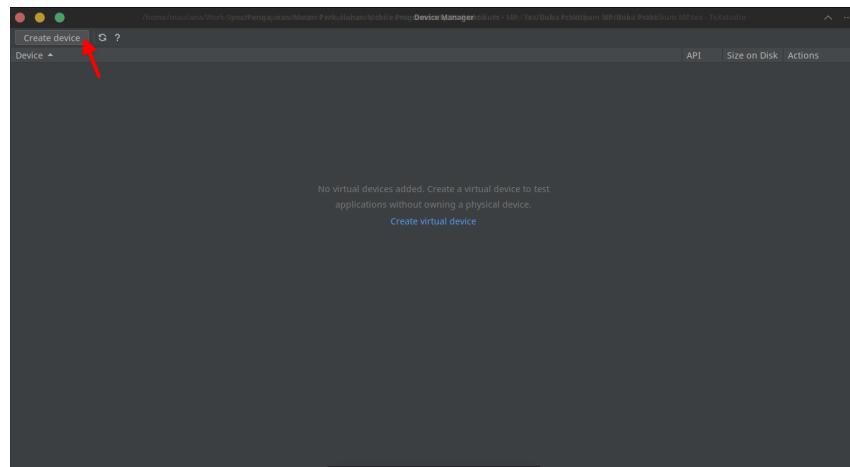
5. Kemudian proses instalasi SDK akan dimulai. Tunggu hingga selesai dan klik **Finish** untuk menutup **SDK Componenet Installer**



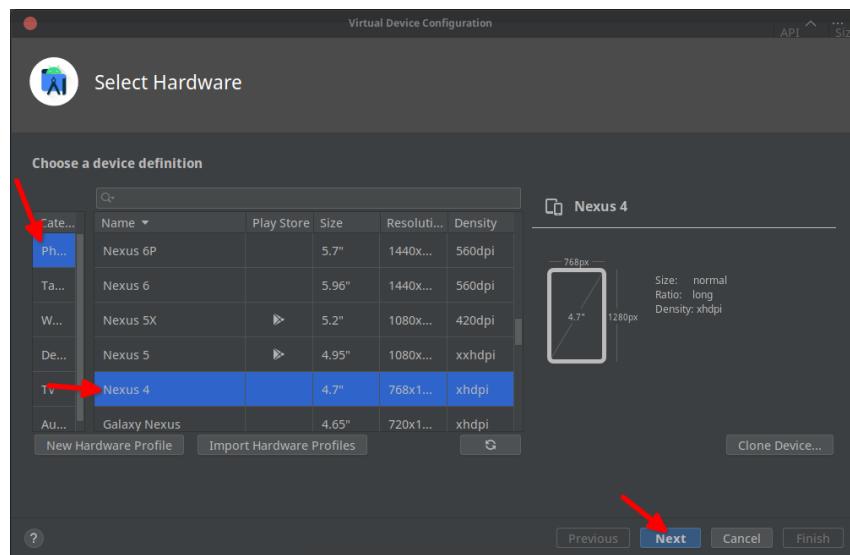
6. Lalu klik **OK** untuk menutup **SDK Manager**
7. Android Studio akan kembali menampilkan halaman **Welcome to Android Studio**. Kemudian konfigurasikan perangkat virtualisasi dengan klik **Titik Tiga / More Action → Virtual Device Manager**



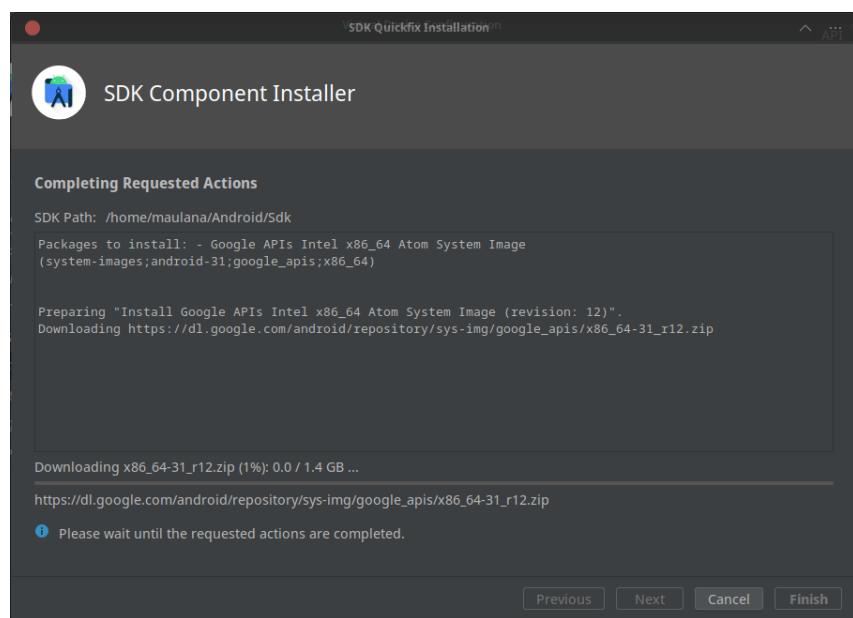
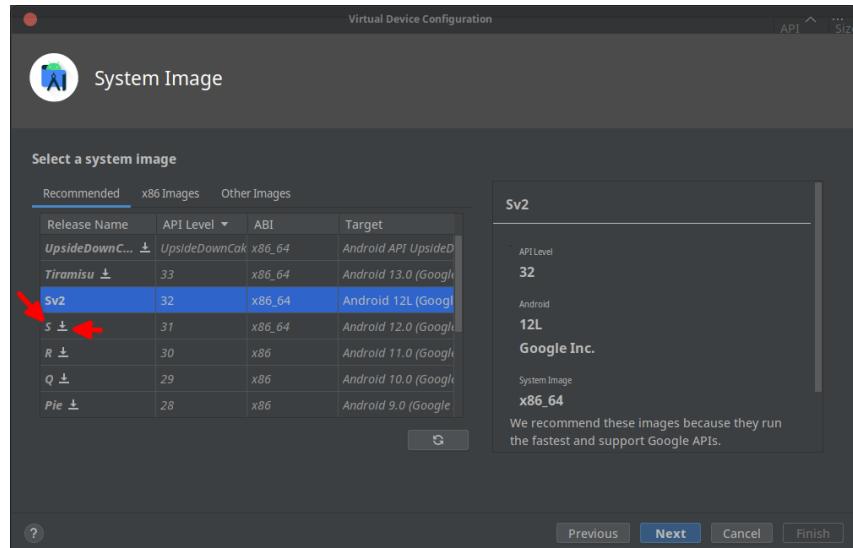
8. Berikutnya akan memunculkan konfigurasi perangkat Virtual. Secara **Default** halaman ini masih kosong, dan bisa memulai membuat dengan meng-klik **Create device**



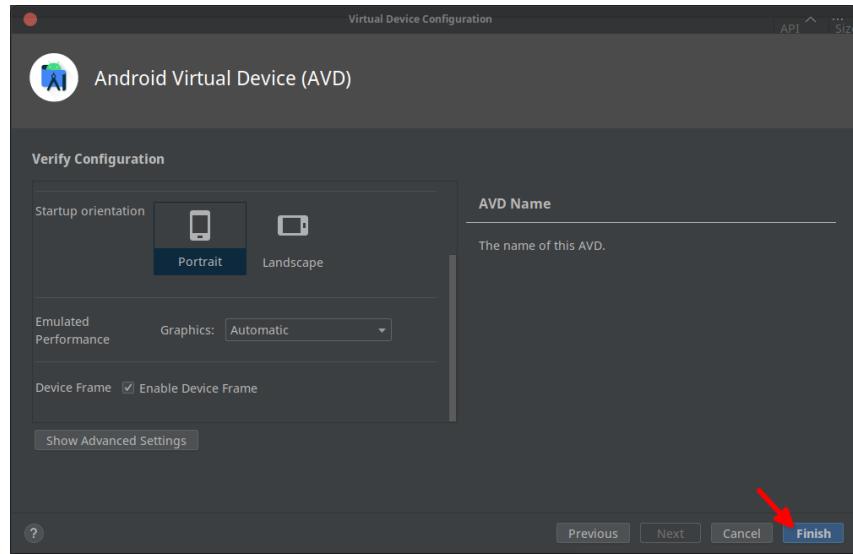
9. Android Studio akan memulai *WIZARD* untuk membuat perangkat **Virtual Android**. Pilih **Phone** → **Nexus 4** → **Next**



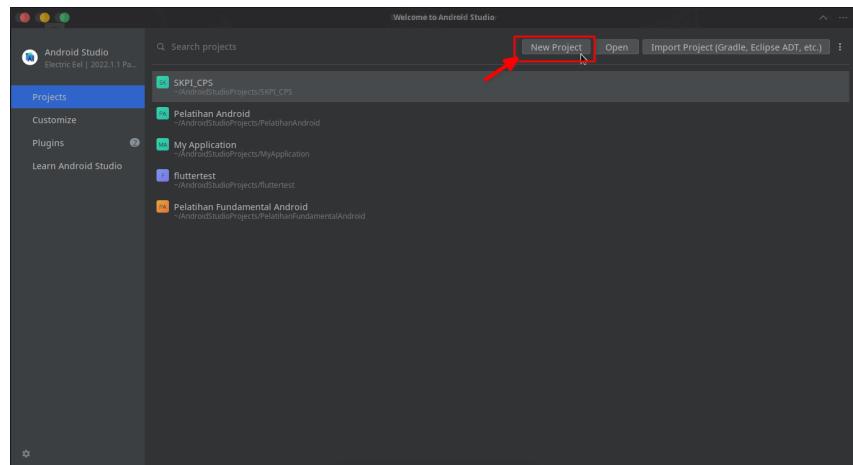
10. Pilih **Versi Android**, → Klik **Icon Download** di samping versi apabila belum terinstall → Klik **Next** apabila **System Image** sudah terinstall.



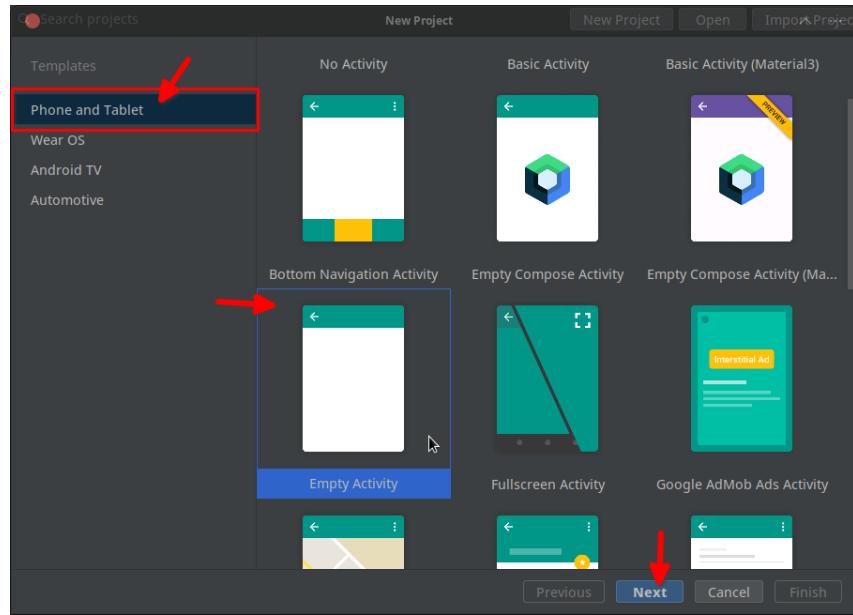
11. Berikutnya adalah konfigurasi perangkat itu sendiri, secara **Default** tidak perlu dirubah kecuali memahami konfigurasi lanjutan dari **Virtual Machine**. Klik **Finish**



12. Jika **Android Studio** meminta untuk menginstall **Intel HAXM**, instal aplikasi tersebut untuk dapat menggunakan **emulator**. Dan pastikan BIOS/UEFI komputer sudah terkonfigurasi untuk **Virtualisasi**
13. Tutup **Virtual Device Manager**, dan Android Studio sudah siap untuk membuat aplikasi. Klik **New Project**.

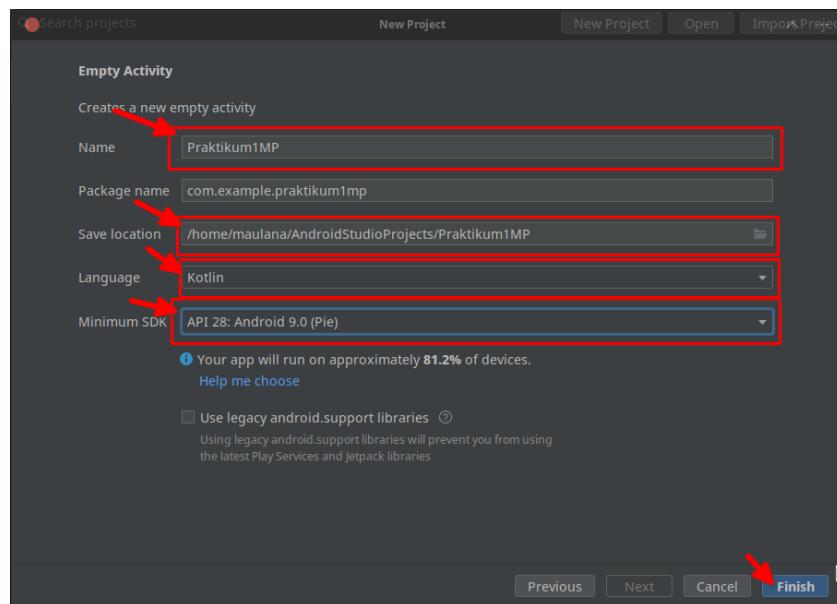


14. Android Studio akan membuka memperlihatkan konfigurasi projek yang akan dibuat. Pilih **Phone and Tablet** → Klik **Empty Activity** => Klik **Next**

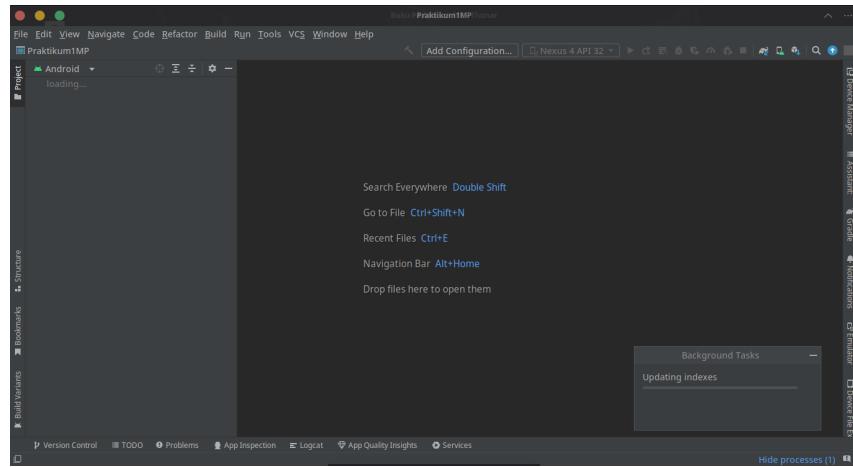


15. Sesuaikan konfigurasi projek seperti berikut, dan klik **Finish**:

- Nama Projek : PrakikumMP_NIM_ANDA
- Lokasi Projek : <Sesuaikan Folder Mahasiswa>
- Bahasa : Kotlin
- Minimum SDK : <Sesuaikan Perangkatl Minimal>

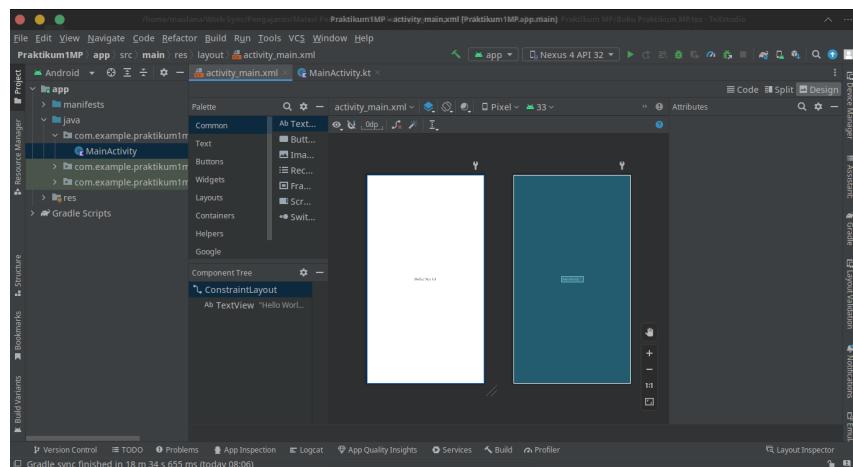
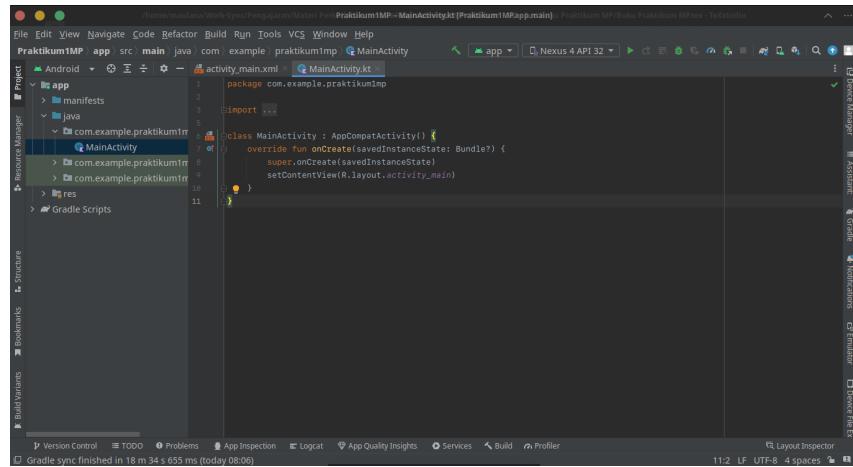


16. Android Studio akan menyiapkan file projek, tunggu hingga selesai. Android Studio akan memperlihatkan kode-kode dan **Project Tree** di samping.

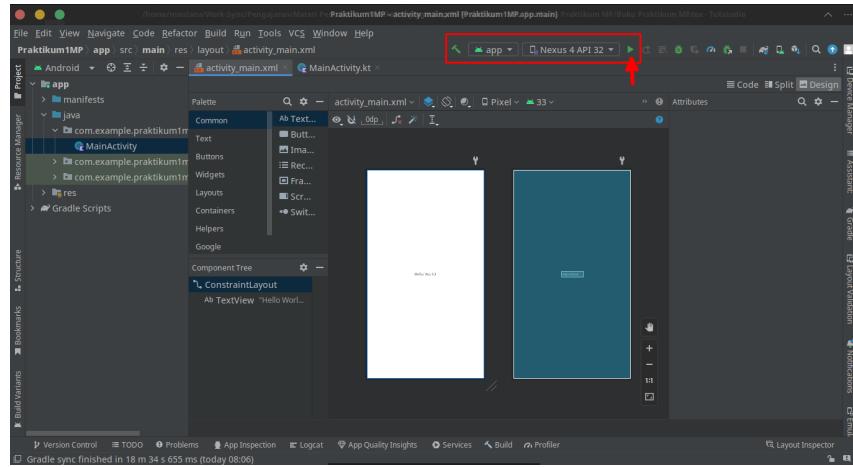


17. Android Studio akan membuat dua (2) file secara **default**:

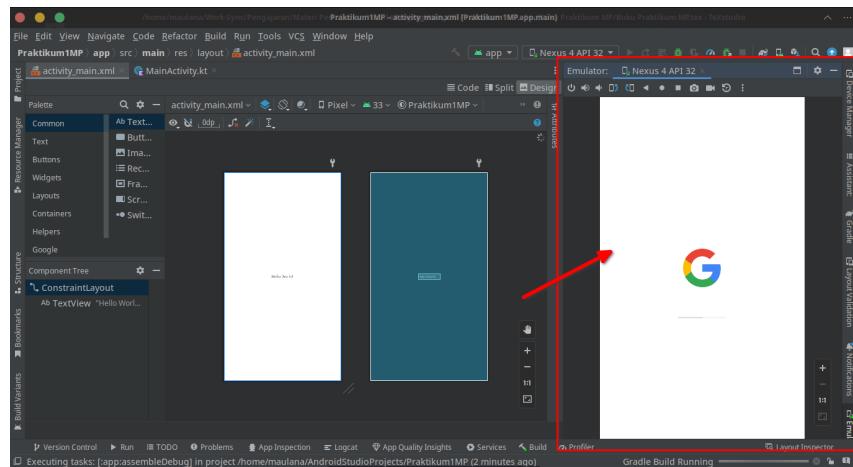
- **MainActivity.kt** : Digunakan untuk menaruh kode-kode fungsional menggunakan bahasa **Java/Kotlin**
- **activity_main.xml** : Digunakan untuk meletakkan objek-objek antarmuka aplikasi Android



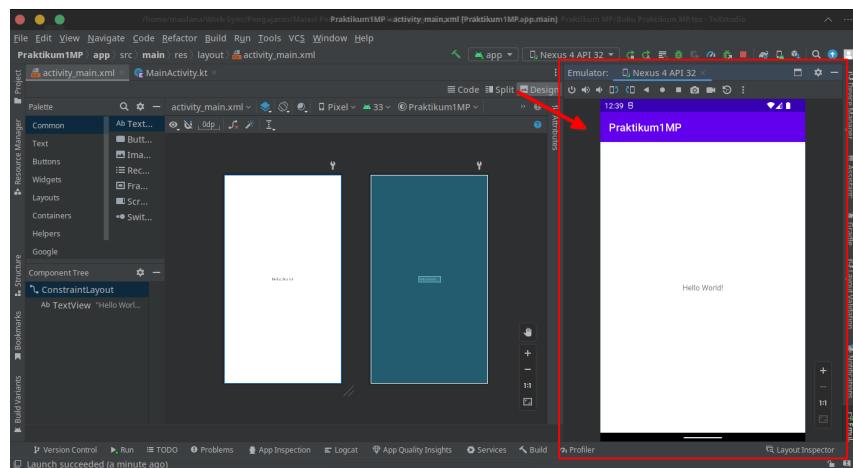
18. Android Studio sudah siap untuk membuat aplikasi Android. Namun sebelum membuat aplikasi, berikutnya adalah mencoba untuk menjalankan aplikasi **Hello World** ini ke **Virtual Device** dengan klik tombol **Play**



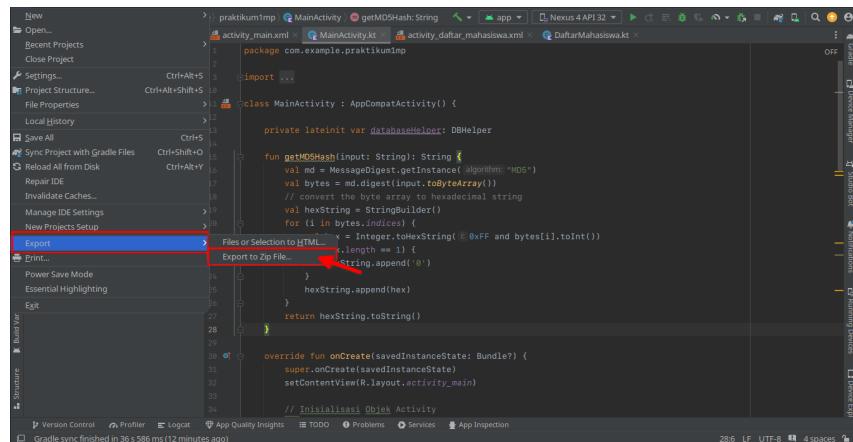
19. Tergantung dari versi Android Studio, versi terbarunya akan membuka **Panel Kanan** untuk menampilkan **Virtual Android**. Tunggu **Kompilasi** dan **Booting** perangkat hingga aplikasi muncul di layar tersebut



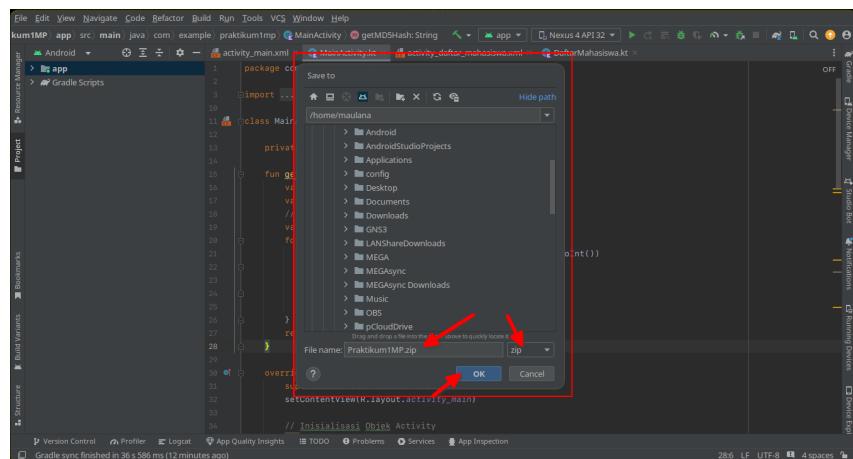
20. Jika sukses, maka Android Studio akan menampilkan aplikasi seperti gambar berikut



21. Untuk mengirimkan hasilnya, klik **File**, pilih **Export**, dan pilih **Export to ZIP**.



22. Pilih lokasi yang mudah diingat, pastikan nama dan format ZIP sudah sesuai, dan klik **OK**



Bab 2

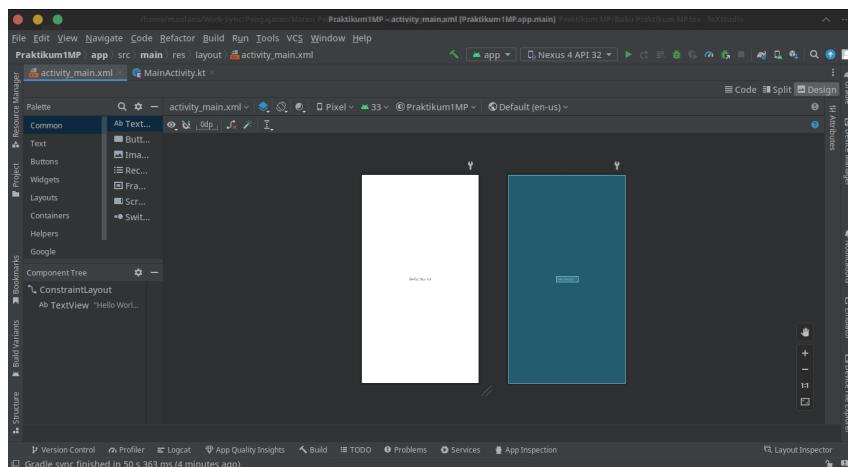
Praktikum 2

2.1 Antarmuka Halaman Login

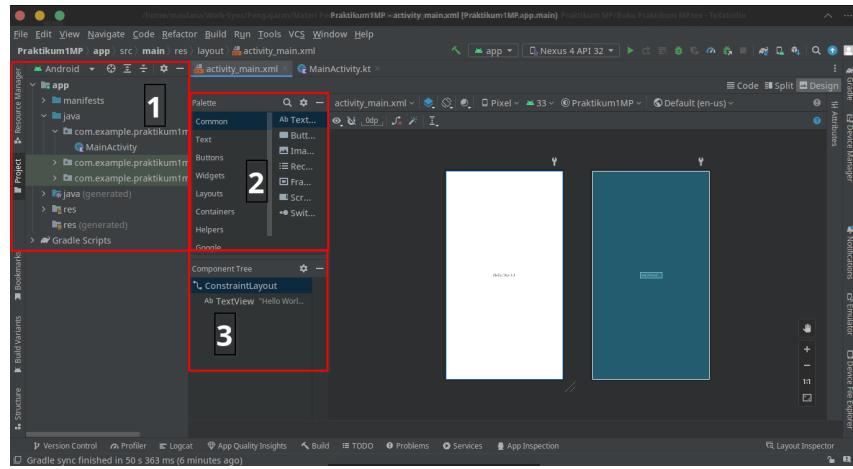
Di bagian ini mahasiswa akan diperkenalkan dengan antarmuka **Android Studio**, sehingga dapat membantu siswa dalam melakukan navigasi yang ada di dalam aplikasi tersebut.

2.2 Tutorial

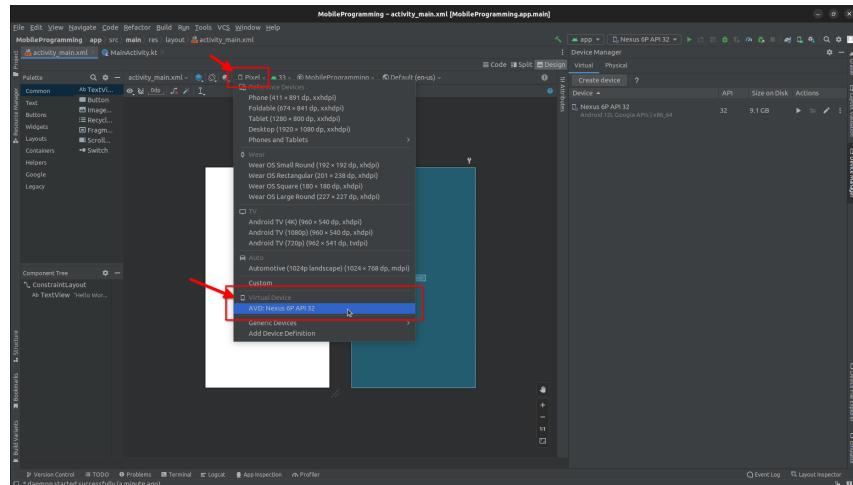
1. Buka **Android Studio**, dan buka kembali **Praktikum1MP** yang sudah dibuat sesi sebelumnya.



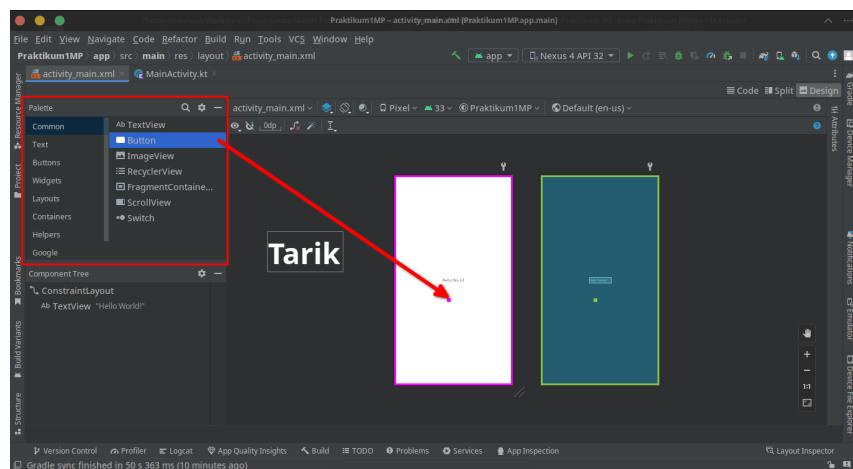
2. Di bagian panel kiri ada dua (2) hal yang tidak boleh dilupakan:
 - (a) Panel **Project** : Menampilkan projek saat ini (bisa dikecilkan)
 - (b) Panel **Palette** : Menampilkan objek yang bisa di tempel
 - (c) Panel **Component Tree** : Menampilkan objek yang sudah di tempel



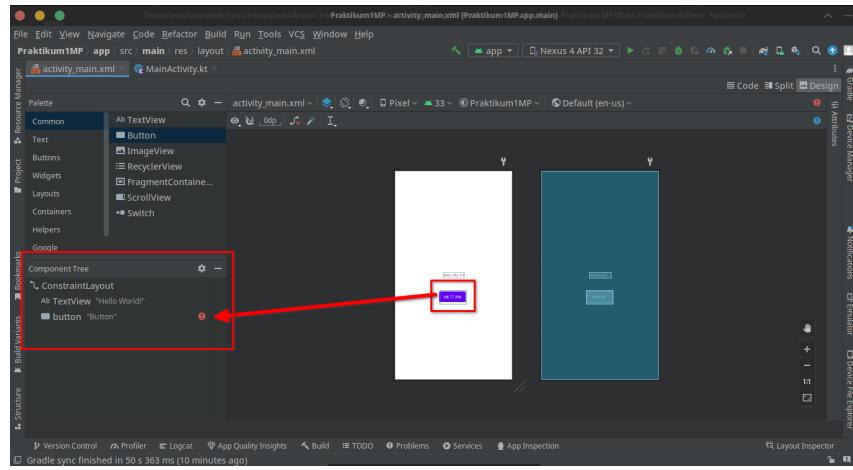
3. Agar **Activity** memiliki ukuran yang sesuai dengan ukuran perangkat, dapat dikonfigurasikan melalui tulisan **Pixel** di atas **Activity**, lalu pilih **Virtual Device: AVD**. Activity akan merubah ukurannya secara otomatis



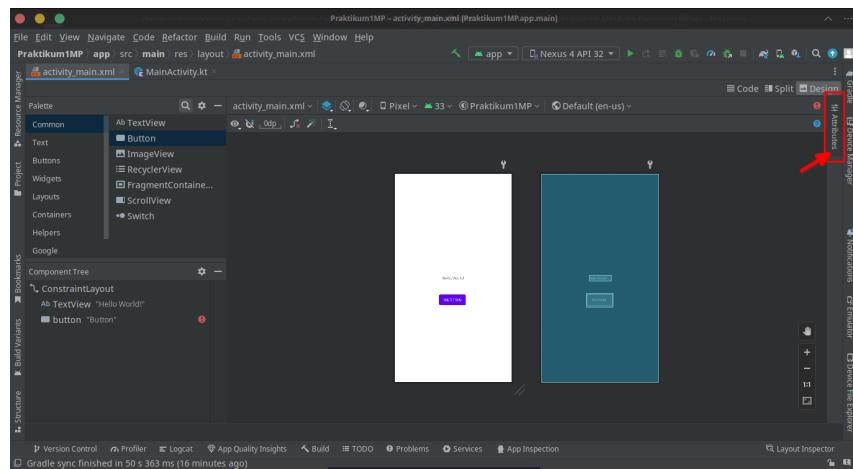
4. Untuk menempelkan objek dari **Palette**, cukup lakukan **Klik dan Tarik** objek ke **Kanvas Putih**. Perhatikan gambar untuk menambahkan Tombol ke **Kanvas**

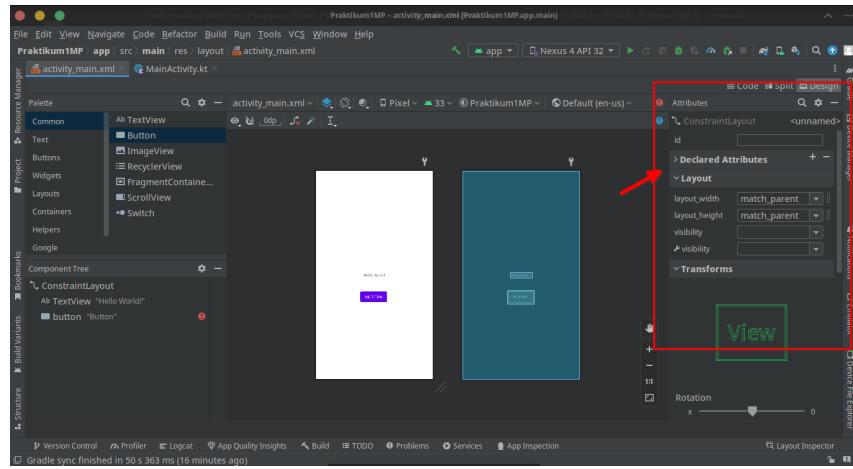


5. Jika dilakukan dengan benar, maka **Tombol** akan terlihat di **Kanvas Putih** dan **Component Tree** di Panel Kiri

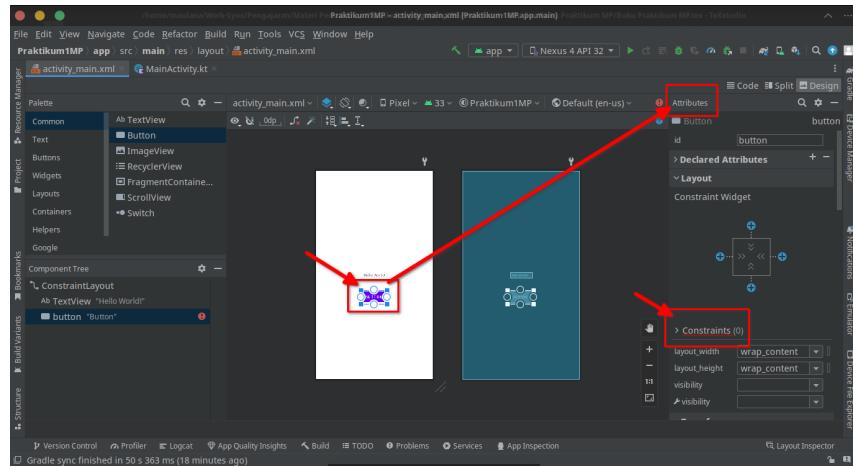


6. Namun objek ini masih memiliki **error** yang terlihat di **Component Tree**. Biasanya error ini muncul karena:
- 1 **Constraint** Vertikal dan 1 **Constraint** Horizontal tidak diatur
 - Ukuran **Tombol** terlalu kecil
7. Karena objek baru ditambahkan, maka **Constraint** belum diatur, untuk mengatur ini klik di Panel Kanan tombol **Attributes**

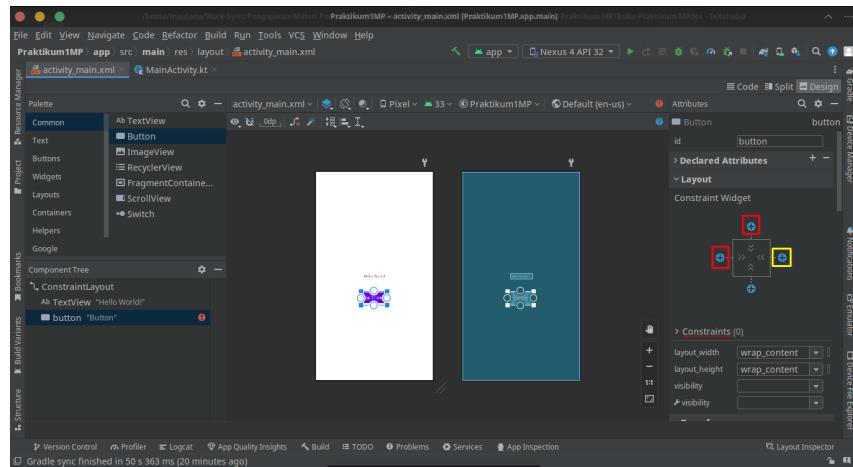




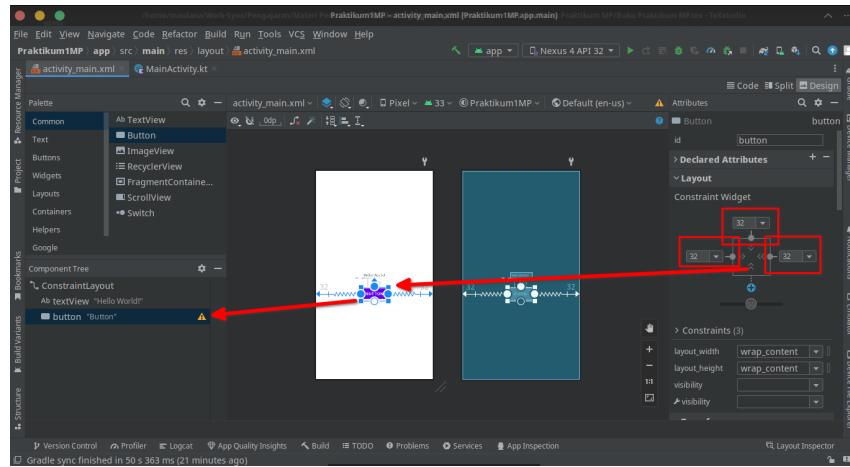
8. Untuk mengkonfigurasikan **Attributes** dari suatu objek, pastikan panel **Attributes** terbuka, dan klik **Objek** yang mau dirubah. Berdasarkan gambar di bawah, **Constraint** tidak di set sehingga muncul indikator Merah



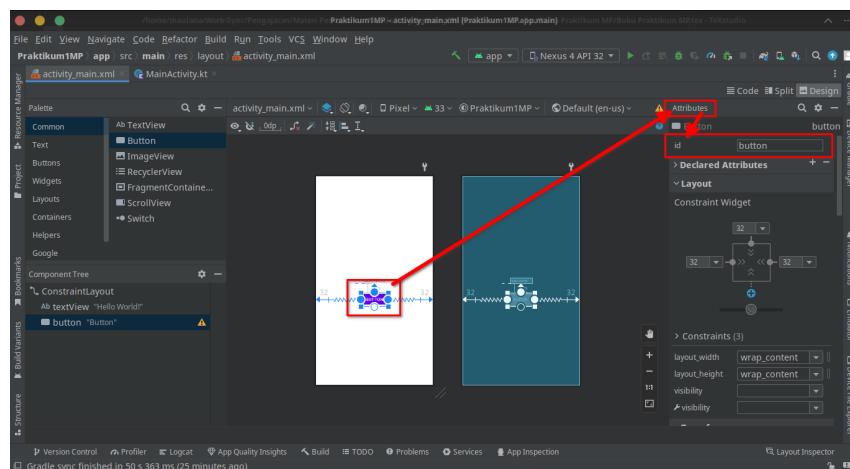
9. Untuk memperbaiki ini, cukup tambahkan **Constraint Atas** dan **Constraint Samping** seperti gambar berikut (Merah Wajib, Kuning Opsional):



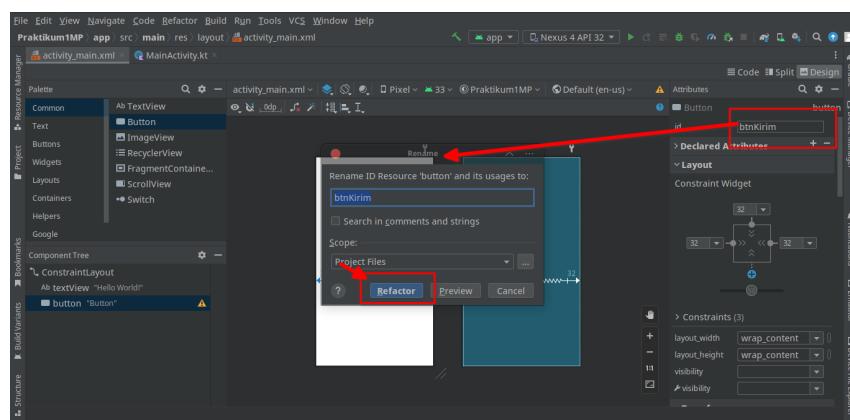
10. Atur jarak **Constraint** secara manual (Contoh 32) jika diperlukan, dan **error** pada objek **Tombol** sudah hilang.



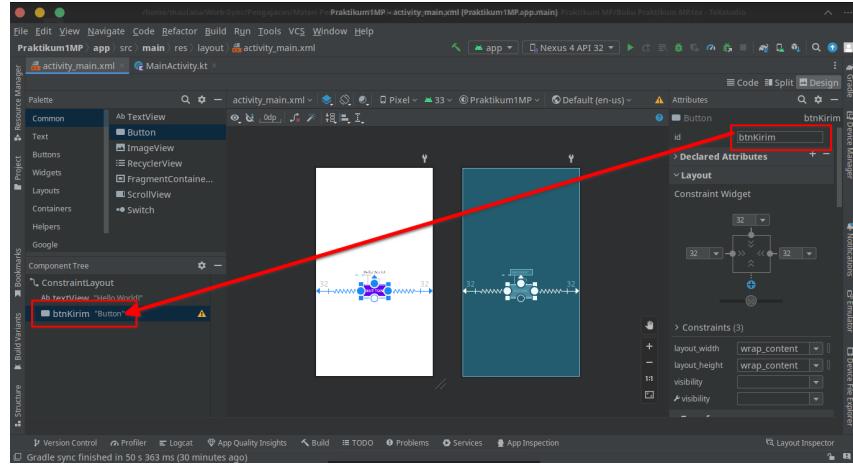
11. Dalam aplikasi Android memiliki banyak kemungkinan untuk ditambahkan objek dalam jumlah yang banyak. Maka untuk membedakan objek satu dengan yang lainnya dapat menggunakan **Attribute ID**
12. Untuk mengatur **ID Objek** cukup klik **Objek** dan lihat panel **Attribute**. Naikkan panel sampai paling atas.



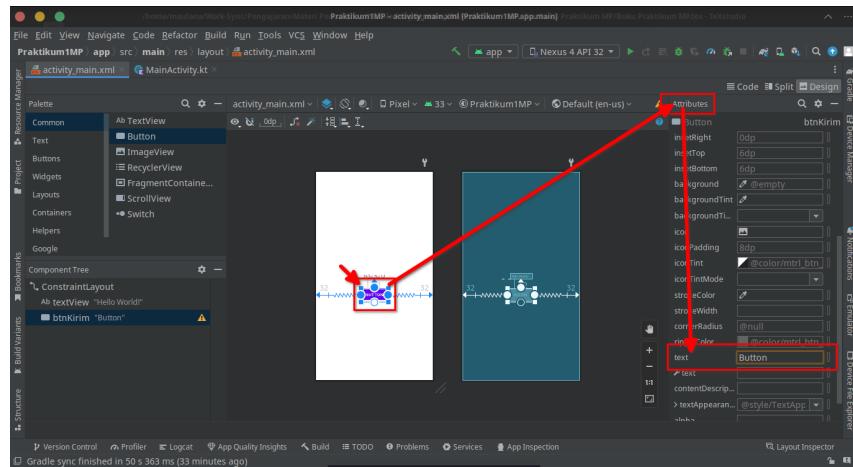
13. Ubah **ID Objek** menjadi **btnKirim** (atau nama unik lainnya). Klik area bebas maka permintaan **Rename** akan muncul. Klik **Refactor** untuk menerapkan peramaan **ID Objek**.



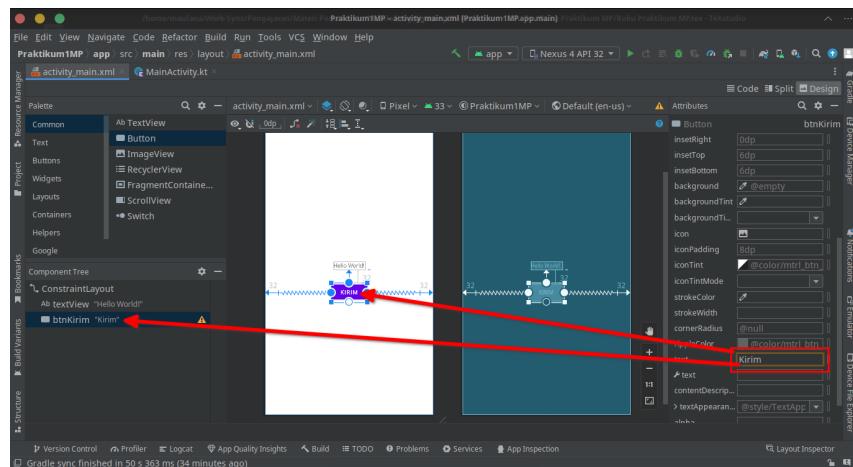
14. Penamaan ini akan langsung berdampak ke **Component Tree**. Maka untuk mengakses **Tombol** ini cukup memanggil ID **Objek** tersebut



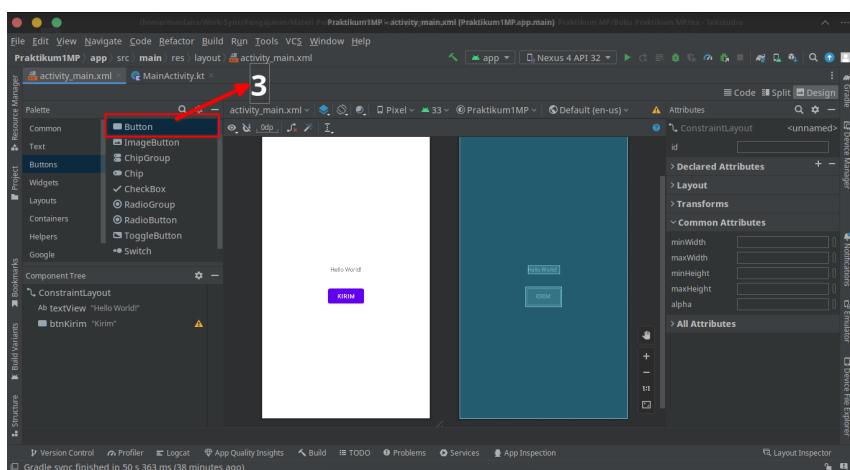
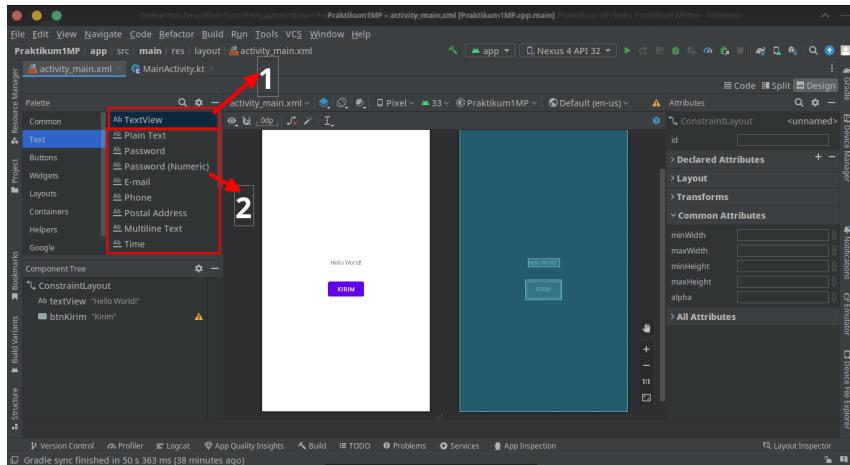
15. Berikutnya adalah mengubah **Isi Teks**. Dengan cara yang sama namun ketika memilih panel **Attribute**, pilih **Text** (Gunakan **Hint** untuk **EditText**)



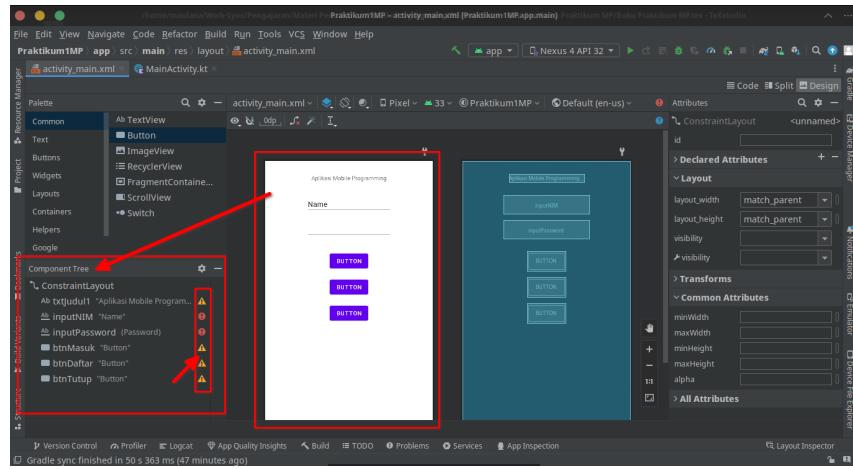
16. Isi langsung dengan teks yang diinginkan lalu Enter



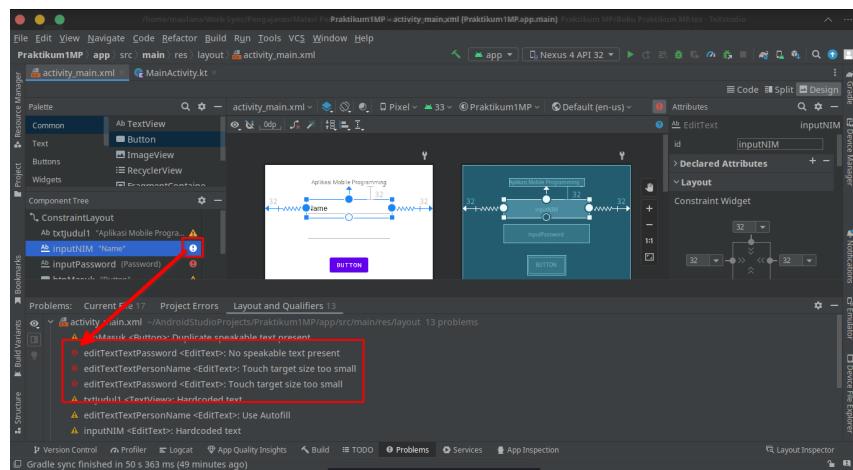
17. Konfigurasi ini berlaku untuk **Semua Objek**. Sehingga memudahkan dalam konfigurasinya.
18. Selain itu ada beberapa objek lain yang wajib diketahui berupa:
 - (a) **TextView** : Digunakan untuk menampilkan teks / String saja tanpa input
 - (b) **EditText** : Digunakan untuk menampilkan dan menerima input dalam bentuk teks / String
 - (c) **Button** : Digunakan untuk pemberi aksi ketika ditekan.



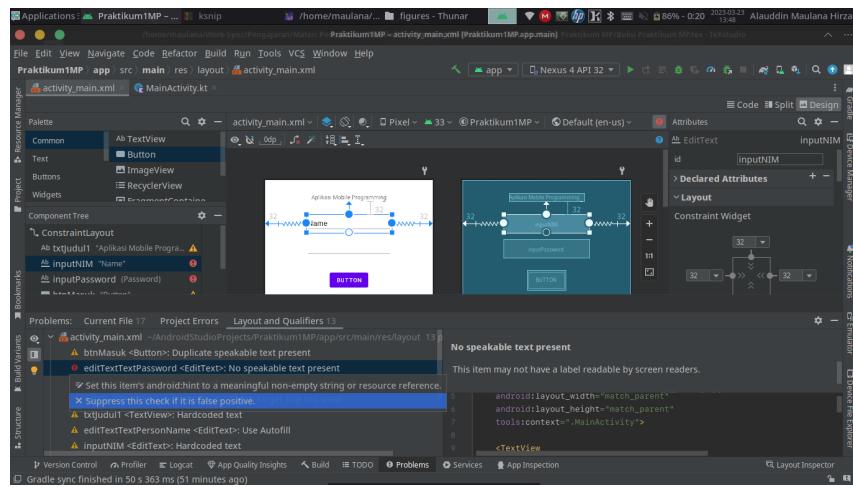
19. Hapus semua objek yang ada di **Kanvas Putih**. Lalu buatlah tampilan seperti berikut dengan menggunakan:
 - Susunlah satu per satu dari atas ke bawah
 - 1 **TextView**
 - 1 **EditText (Plain Text)**
 - 1 **EditText (Password)**
 - 3 **Button**
 - Atur **Constraint** SEMUA OBJEK di atas seperti Langkah 9 dan 10



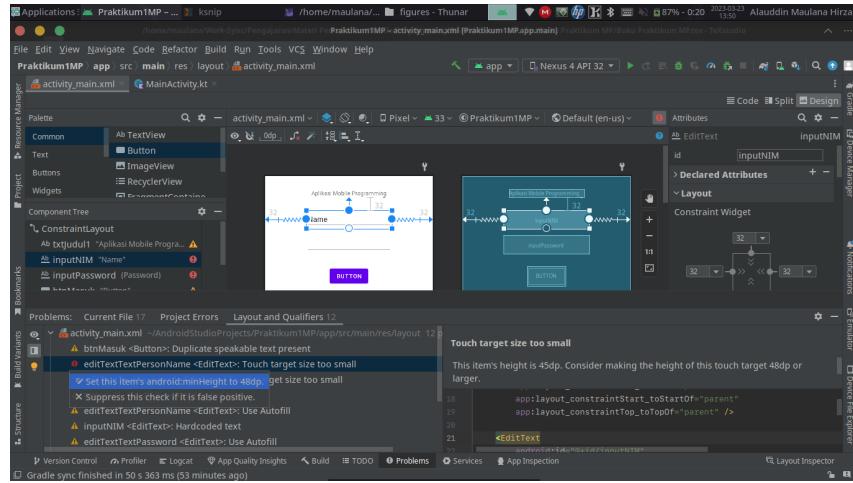
20. Ketika semua terpasang, maka akan muncul khususnya untuk **EditText** nya. Klik **Indikator Error Merah** untuk menampilkan **error**.



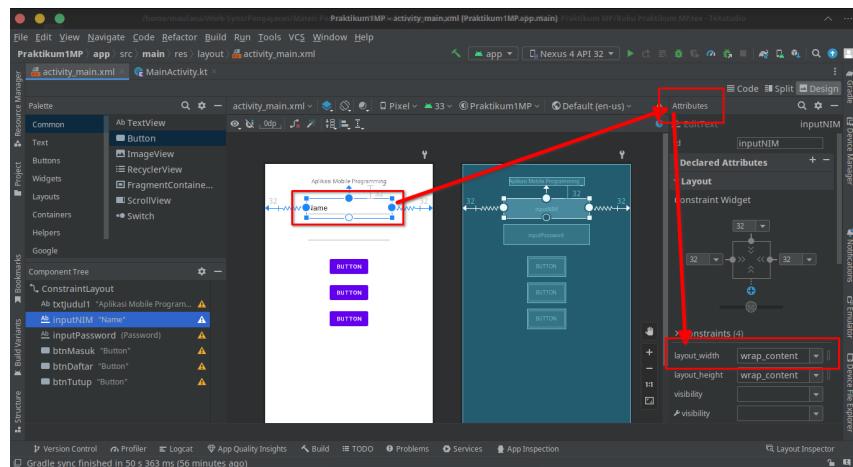
21. Untuk memperbaiki ini, gunakan **Klik Kanan**, pilih **Show Quick Fixes** lalu pilih:
- Jika **Error No speakable text present** → **SUPPRESS THIS CHECK IF IT IS FALSE POSITIVE**



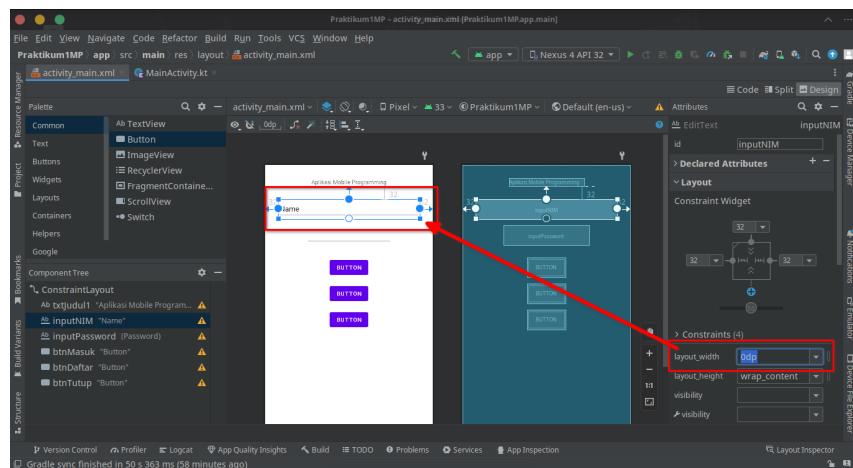
- Jika Error Touch target size too small → Set this item's android:minHeight to 48dp



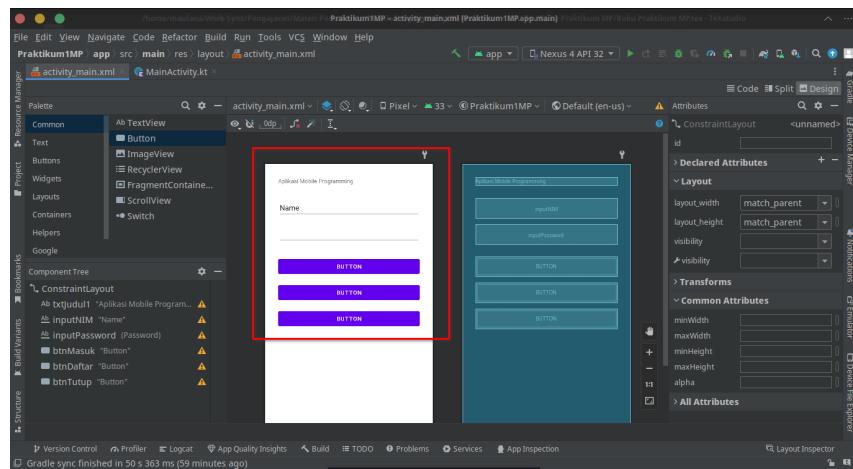
22. Error akan hilang. Namun tampilan terlalu jelek. Untuk mengatur **Lebar Objek** cukup klik **Object** → **Attribute** → **layout_width**



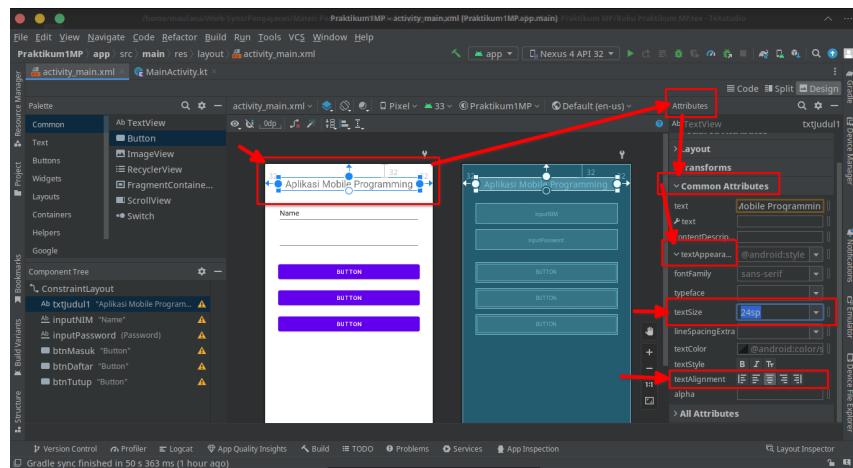
23. Ubah dari **wrap_content** ke **0dp** atau **match_constraint** untuk melebarkan objek selebar **constraint**



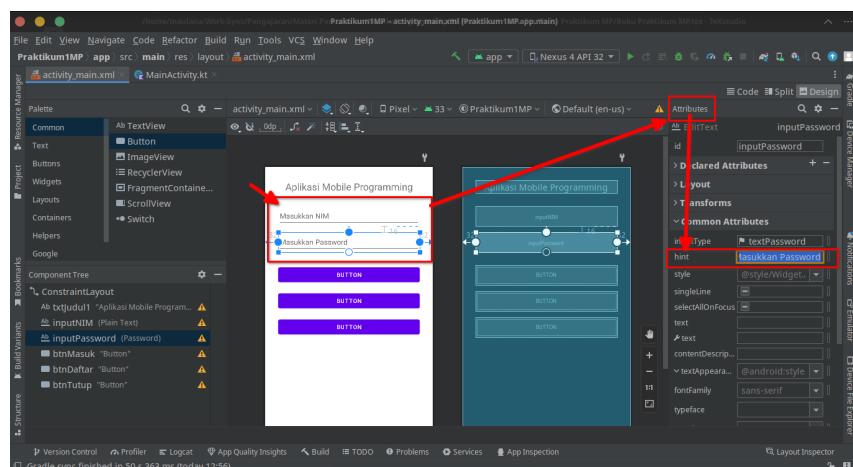
24. Ulangi ke **Semua Objek** hingga terlihat seperti berikut



25. Ubah Letak Teks dari **TextView** Judul menjadi **Center** dan berukuran **24sp**. Klik Objek → Attribute → Common Attributes → textappearances

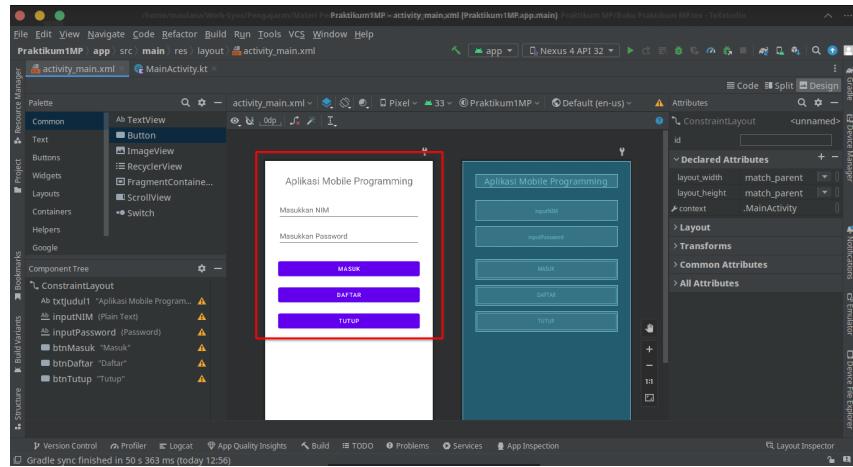


26. Ubah Hint **EditText** (**inputNIM**) dan **EditText** (**inputPassword**) menjadi seperti berikut (Pastikan **Attribute Text Kosong**):



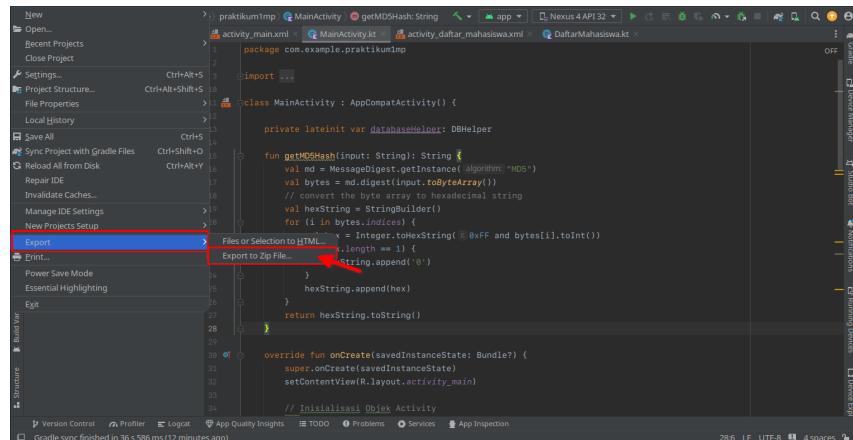
27. Terakhir, ubah **Teks** dari masing-masing **Tombol** menjadi:

- btnMasuk → Masuk
- btnDaftar → Daftar
- btnTututp → Tutup

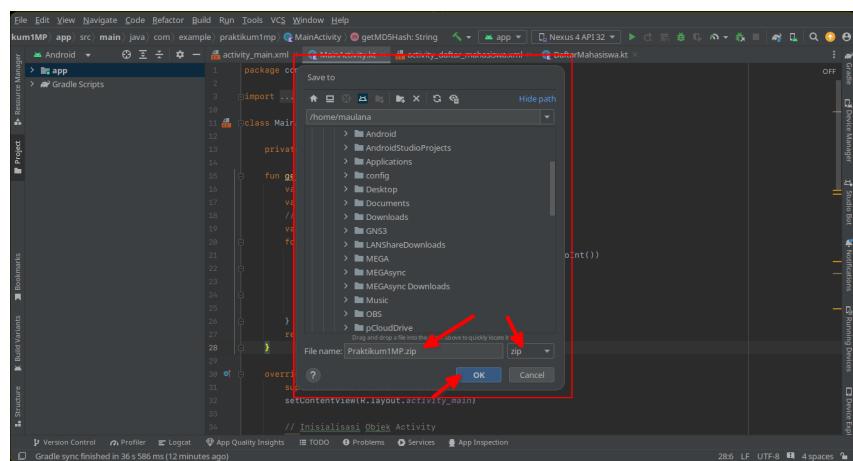


28. Screenshot hasil dan kirimkan ke **E-Learning**

29. Untuk mengirimkan hasilnya, klik **File**, pilih **Export**, dan pilih **Export to ZIP**.



30. Pilih lokasi yang mudah diingat, pastikan nama dan format ZIP sudah sesuai, dan klik **OK**



Bab 3

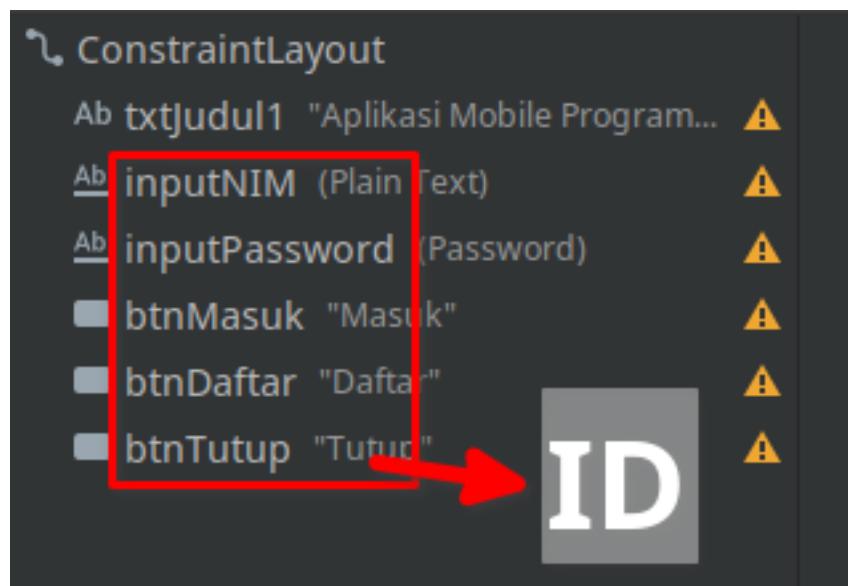
Praktikum 3

3.1 Pemrograman Halaman Login

Di bagian ini mahasiswa akan diperkenalkan dengan antarmuka **Android Studio**, sehingga dapat membantu siswa dalam melakukan navigasi yang ada di dalam aplikasi tersebut.

3.2 Tutorial

1. Untuk memulai memrogram objek-objek yang ada di dalam Activity, buka lah proyek sebelumnya.
2. Buka **activity_main.xml**, dan pastikan ID dari masing-masing objek sudah diatur



3. Setelah semua objek sudah memiliki ID (Kecuali Judul), berikutnya buka file **MainActivity.kt** dan tambahkan ruang kosong untuk memasukkan kode fungsional

```
activity_main.xml < MainActivity.kt
1 package com.example.praktikumimp
2
3 import ...
4
5
6 class MainActivity : AppCompatActivity() {
7     override fun onCreate(savedInstanceState: Bundle?) {
8         super.onCreate(savedInstanceState)
9         setContentView(R.layout.activity_main)
10    }
11 }
```

4. Masukkan kode berikut untuk menginisialisasi / mengubah objek menjadi variabel kode

Potongan Kode

```
// Inisialisasi Objek Activity  
var inputNIM = findViewById<EditText>(R.id.inputNIM)  
var inputPassword = findViewById<EditText>(R.id.inputPassword)  
val btnMasuk = findViewById<Button>(R.id.btnMasuk)  
val btnDaftar = findViewById<Button>(R.id.btnDaftar)  
val btnTutup = findViewById<Button>(R.id.btnTutup)
```

5. Namun kode ini memiliki **error** cukup **Arahkan** mouse dan **Klik** baris kode yang error, Android Studio akan menawarkan untuk solusi permasalahan. Lalu tekan **Alt+Enter** sesuai arahan rekomendasi itu.

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
praktikum1MP app src main java com example praktikum1mp MainActivity onCreate
activity_main.xml MainActivity.kt

1 package com.example.praktikum1mp
2
3 import ...
4
5
6 class MainActivity : AppCompatActivity() {
7     override fun onCreate(savedInstanceState: Bundle?) {
8         super.onCreate(savedInstanceState)
9         setContentView(R.layout.activity_main)
10        // Inisialisasi Objek Activity
11        var inputTLM = findViewById<EditText>(R.id.inputTLM)
12        var inputPassword = findViewById<EditText>(R.id.inputPassword)
13        val btnMasuk = findViewById<Button>(R.id.btnMasuk)
14        val btnDaftar = findViewById<Button>(R.id.btnDaftar)
15        val btnTutup = findViewById<Button>(R.id.btnTutup)
16    }
17
18 }
19
```

6. Ulangi ke semua kode sehingga tidak muncul Error

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
praktikum1MP app src main java com example praktikum1mp MainActivity onCreate
activity_main.xml MainActivity.kt

1 package com.example.praktikum1mp
2
3 import ...
4
5
6 class MainActivity : AppCompatActivity() {
7     override fun onCreate(savedInstanceState: Bundle?) {
8         super.onCreate(savedInstanceState)
9         setContentView(R.layout.activity_main)
10        // Inisialisasi Objek Activity
11        var inputTLM = findViewById<EditText>(R.id.inputTLM)
12        var inputPassword = findViewById<EditText>(R.id.inputPassword)
13        val btnMasuk = findViewById<Button>(R.id.btnMasuk)
14        val btnDaftar = findViewById<Button>(R.id.btnDaftar)
15        val btnTutup = findViewById<Button>(R.id.btnTutup)
16    }
17
18 }
19
```

7. Jika sudah langkah berikutnya adalah membuat kode Aksi untuk **Tombol/Button**. Masukkan kode berikut persis di bawah kode **Inisialisasi** sebelumnya. Ruang kosong di dalam kode adalah disengaja

Potongan Kode

```
btnMasuk.setOnClickListener {
}
btnDaftar.setOnClickListener {
}
btnTutup.setOnClickListener {
}
```

```

file Edit View Navigate Code Refactor Build Run Tools VCS Window Help
PraktikumIMP app src main java com.example.praktikumIMP MainActivity
activity_main.xml x MainActivity.kt ...
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    // Inisialisasi Objek Activity
    var inputNIM = findViewById<EditText>(R.id.inputNIM)
    var inputPassword = findViewById<EditText>(R.id.inputPassword)
    val btnMasuk = findViewById<Button>(R.id.btnMasuk)
    val btnDaftar = findViewById<Button>(R.id.btnDaftar)
    val btnTutup = findViewById<Button>(R.id.btnTutup)

    btnMasuk.setOnClickListener { view ->
        }
    btnDaftar.setOnClickListener { view ->
        }
    btnTutup.setOnClickListener { view ->
        }
}

```

8. Berikutnya mengisikan fungsi-fungsi di masing-masing **Tombol**. Di mulai dari **btnTutup**. Masukkan kode berikut:

Potongan Kode

```
// Perintahkan Activity untuk Selesai
finish()
```

```

file Edit View Navigate Code Refactor Build Run Tools VCS Window Help
PraktikumIMP app src main java com.example.praktikumIMP MainActivity
activity_main.xml x MainActivity.kt ...
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Inisialisasi Objek Activity
        var inputNIM = findViewById<EditText>(R.id.inputNIM)
        var inputPassword = findViewById<EditText>(R.id.inputPassword)
        val btnMasuk = findViewById<Button>(R.id.btnMasuk)
        val btnDaftar = findViewById<Button>(R.id.btnDaftar)
        val btnTutup = findViewById<Button>(R.id.btnTutup)

        btnMasuk.setOnClickListener { view ->
            }
        btnDaftar.setOnClickListener { view ->
            }
        btnTutup.setOnClickListener { view -
            // Perintahkan Activity untuk Selesai
            finish()
        }
    }
}

```

9. Kode berikutnya adalah untuk mengambil data login dari **EditText** yang ditempelkan di **activity_main.xml**. Namun saat ini kode hanya bisa menampilkan saja.

Potongan Kode

```

val dataNIM:String = inputNIM.text.toString()
val dataPassword:String = inputPassword.text.toString()
// Tampilkan Hasil
Toast.makeText(applicationContext,dataNIM+" "+dataPassword,Toast.LENGTH_SHORT)
.show()
```

The screenshot shows the Android Studio interface. On the left, the code editor displays `MainActivity.kt` with several lines of Kotlin code. A red box highlights the following section of code:

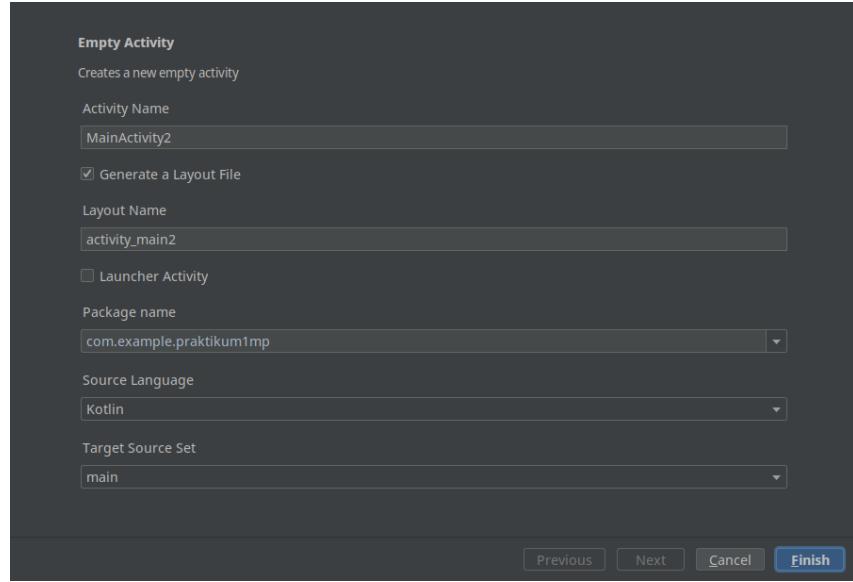
```

        btnMasuk.setOnClickListener { view ->
            val dataNIM = inputNIM.text.toString()
            val dataPassword = inputPassword.text.toString()
            // Tampilkan Hasil
            Toast.makeText(getApplicationContext(), "dataNIM=" + dataNIM + "dataPassword", Toast.LENGTH_SHORT)
                .show()
        }
        btnTutup.setOnClickListener { view ->
            // Perintahkan Activity untuk Selesai
            finish()
        }
    }
}

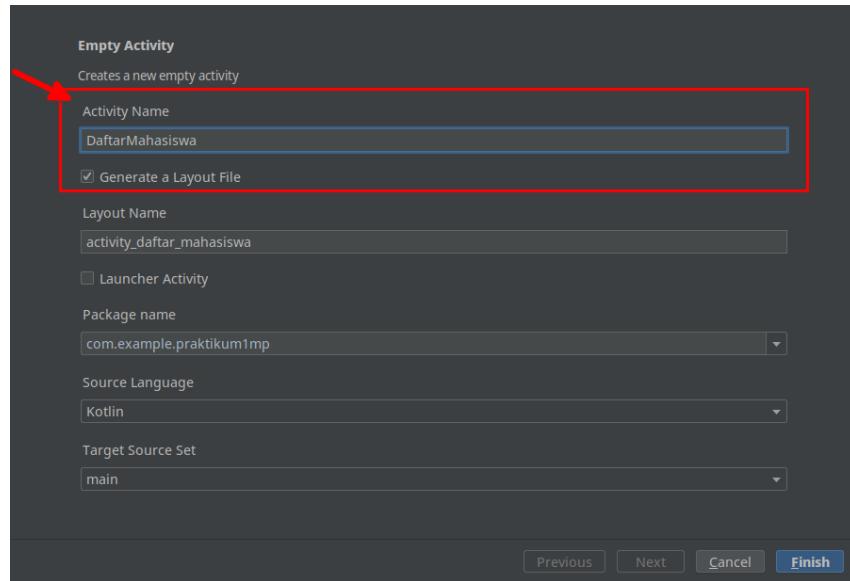
```

On the right, an emulator window shows a dark-themed Android interface with various app icons.

- Sebelum memasukkan kode terakhir, buatlah **Activity** baru dengan cara: Buka Panel Project → Klik Kanan Folder **app** → Pilih **new** → Pilih **Activity** → Pilih **Empty Activity**. Maka muncul Window untuk membuat **Activity** baru



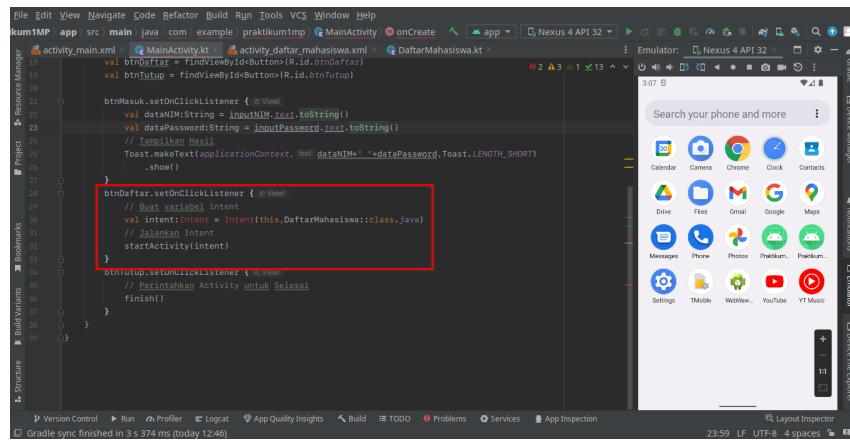
- Ubah **MainActivity2** menjadi **DaftarMahasiswa** di bagian **Activity Name**. Sisanya akan mengikuti. Lalu klik **Finish**



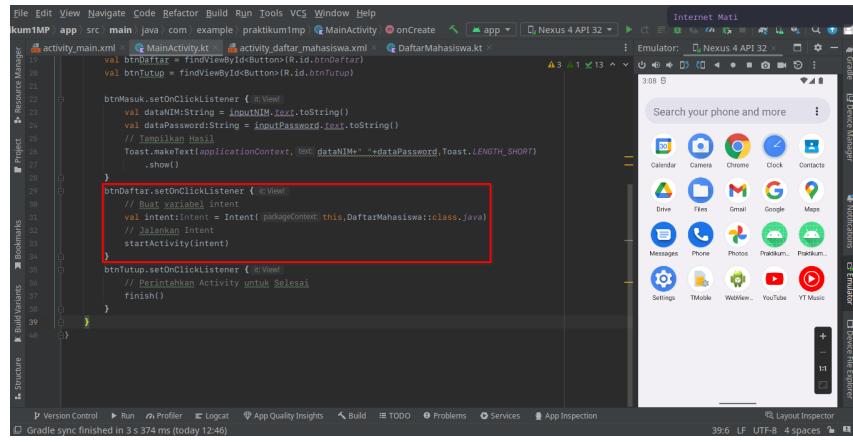
12. Android Studio akan menampilkan layout dan kode kotlin **Activity** baru ini. Kembali buka **MainActivity.kt**
13. Masukkan kode baru untuk tombol **btnDaftar** dengan kode seperti berikut:

Potongan Kode

```
// Buat variabel intent
val intent:Intent = Intent(this,DaftarMahasiswa::class.java)
// Jalankan Intent
startActivity(intent)
```



14. Kode akan mengalami **Error** di bagian Intent, gunakan **Alt+Enter** untuk menyelesaikan masalah **import**



15. Tes aplikasi dengan menjalankannya ke Emulator
16. Jika sudah berhasil, tambahkan objek-objek ke layout `activity_daftar_mahasiswa.xml` seperti berikut:

- **TextView**

- **ID** : `txtJudulDaftar`
- **Text** : **Pendaftaran Data Mahasiswa**

- **EditText (Plain)**

- **ID** : `inputDaftarNIM`
- **Hint** : **Masukkan NIM**

- **EditText (Plain)**

- **ID** : `inputDaftarNama`
- **Hint** : **Masukkan Nama**

- **EditText (Email)**

- **ID** : `inputDaftarEmail`
- **Hint** : **Masukkan E-Mail**

- **EditText (Password)**

- **ID** : `inputDaftarPassword`
- **Hint** : **Masukkan Password**

- **Button**

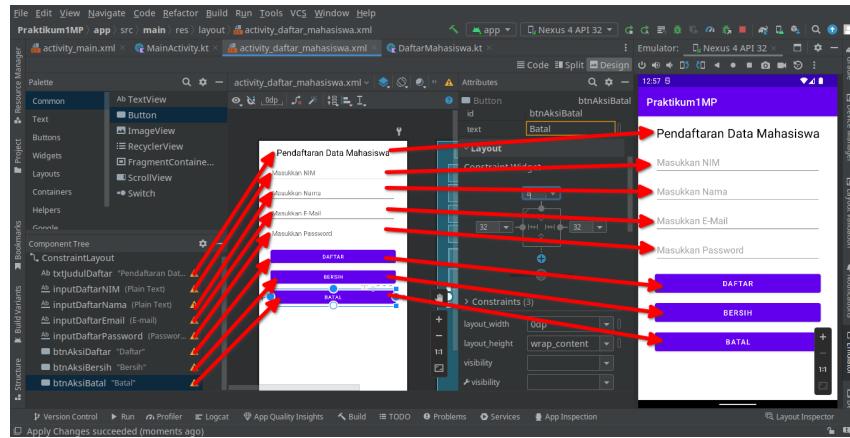
- **ID** : `btnAksiDaftar`
- **Text** : **Daftar**

- **Button**

- **ID** : `btnAksiBersih`
- **Text** : **Bersih**

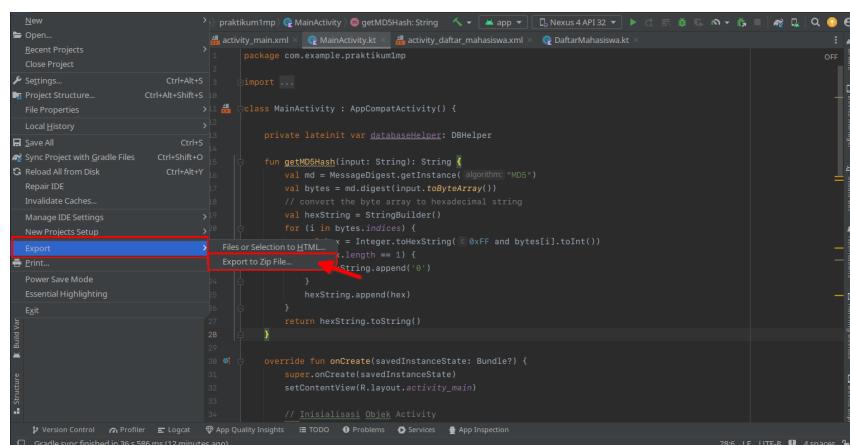
- **Button**

- **ID : btnAksiBatal**
- **Text : Batal**

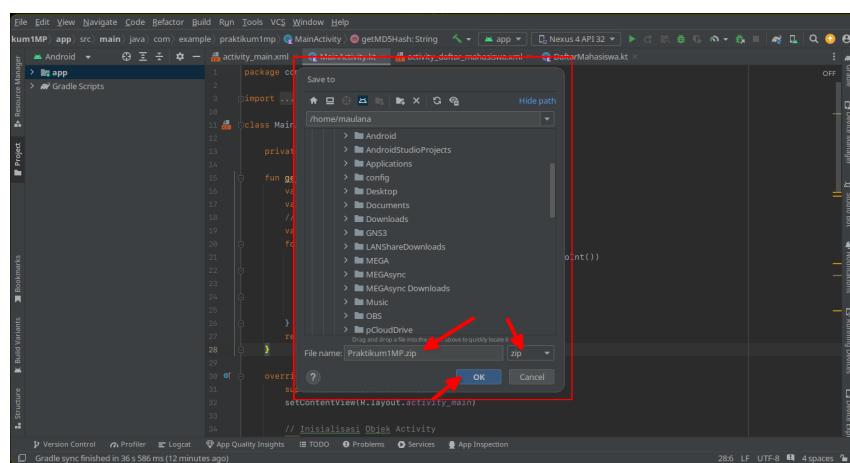


17. Jalan kan aplikasi, dan untuk berpindah ke halaman berikutnya klik tombol **Daftar**
Screenshot hasil dan kirimkan ke **E-Learning**

18. Untuk mengirimkan hasilnya, klik **File**, pilih **Export**, dan pilih **Export to ZIP**.



19. Pilih lokasi yang mudah diingat, pastikan nama dan format ZIP sudah sesuai, dan klik **OK**



Bab 4

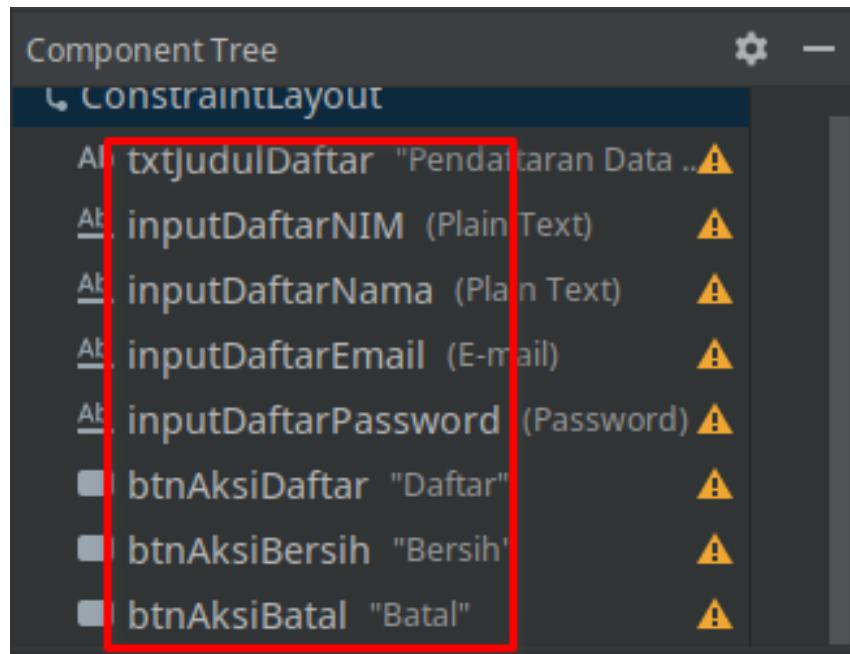
Praktikum 4

4.1 Pemrograman Halaman Pendaftaran

Di bagian ini mahasiswa melanjutkan projek sebelumnya dengan menambahkan kode aksi di halaman Pendaftaran. Mahasiswa akan diperkenalkan bagaimana melakukan akses data dan menampilkan data.

4.2 Tutorial

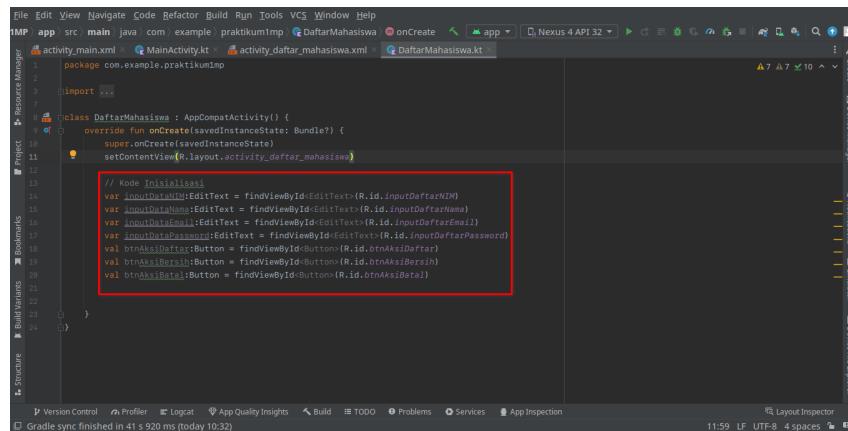
1. Buka projek sebelumnya yang sudah memiliki **Activity** Pendaftaran
2. Pastikan **ID** yang ada di dalam file **activity_daftar_mahasiswa.xml** persis dengan gambar berikut:



3. Jika **ID** pada layout sudah sesuai, maka buka file **DaftarMahasiswa.kt** dan masukkan kode inisialisasi tepat di bawah baris **setContentView** seperti berikut:

Potongan Kode

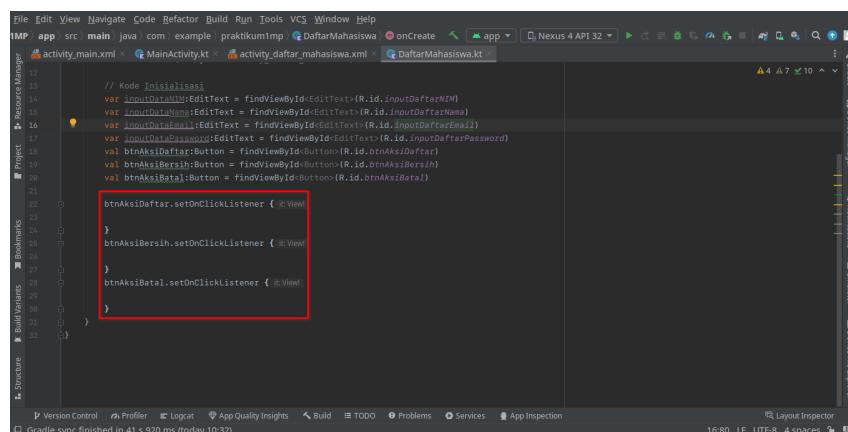
```
// Kode Inisialisasi  
var inputDataNIM:EditText = findViewById<EditText>(R.id.inputDaftarNIM)  
var inputDataNama:EditText = findViewById<EditText>(R.id.inputDaftarNama)  
var inputDataEmail:EditText = findViewById<EditText>(R.id.inputDaftarEmail)  
var inputDataPassword:EditText = findViewById<EditText>(R.id.inputDaftarPassword)  
val btnAksiDaftar:Button = findViewById<Button>(R.id.btnAksiDaftar)  
val btnAksiBersih:Button = findViewById<Button>(R.id.btnAksiBersih)  
val btnAksiBatal:Button = findViewById<Button>(R.id.btnAksiBatal)
```



4. Perbaiki **Error** karena **Import** sebelum melanjutkan ke tahap berikutnya
5. Masukkan kode **Event Listener** untuk masing-masing tombol dengan kode seperti berikut:

Potongan Kode

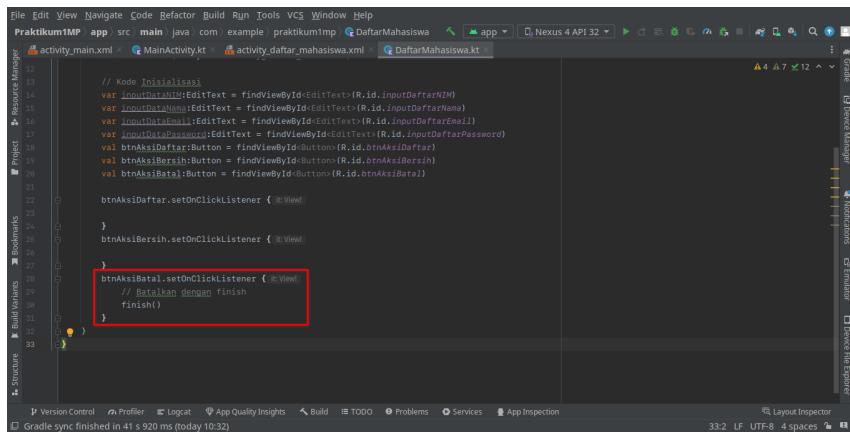
```
btnAksiDaftar.setOnClickListener {  
  
}  
btnAksiBersih.setOnClickListener {  
  
}  
btnAksiBatal.setOnClickListener {  
  
}
```



6. Agar tombol-tombol bisa melakukan aksinya, masukkan kode berikut mulai dari tombol **btnAksiBatal**

Potongan Kode

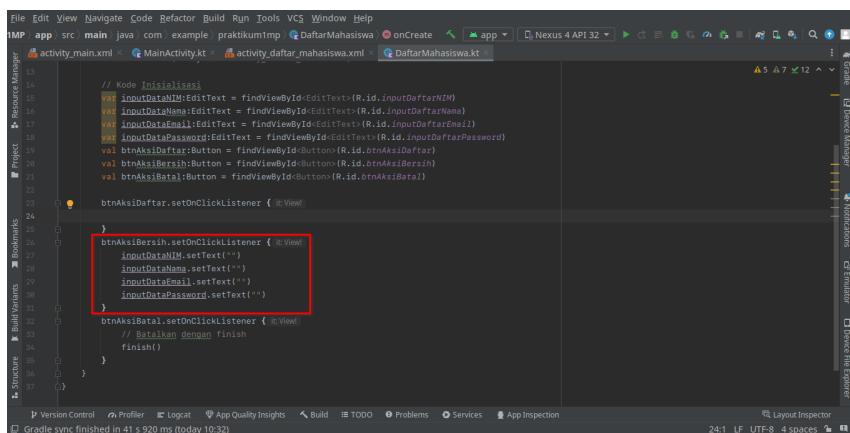
finish()



7. Kemudian masukkan kode untuk membersihkan input di **Daftar Mahasiswa**. Kode ini untuk **btnAksiBersih**

Potongan Kode

```
inputDataNIM.setText("")  
inputDataNama.setText("")  
inputDataEmail.setText("")  
inputDataPassword.setText("")
```



8. Kode berikutnya digunakan untuk mengambil data dari input, lalu mengamankan password agar tidak menggunakan teks biasa / **plaintext**. Perhatikan kode berikut:

Potongan Kode

```
// Ambil Data dari Edit Text
val NIM:String = inputDataNIM.text.toString()
val Nama:String = inputDataNama.text.toString()
val Email:String = inputDataEmail.text.toString()
val Password:String = inputDataPassword.text.toString()
```

```

    val btnAkisDaftar:Button = findViewById(R.id.btnAkisDaftar)
    val btnAkisBersih:Button = findViewById(R.id.btnAkisBersih)
    val btnAkisBatal:Button = findViewById(R.id.btnAkisBatal)

    btnAkisDaftar.setOnClickListener { e:View ->
        // Ambil Data dari Edit Text
        val NIMString = inputDataNIM.text.toString()
        val NameString = inputDataNama.text.toString()
        val EmailString = inputDataEmail.text.toString()
        val PasswordString = inputDataPassword.text.toString()
    }

```

9. Jika sudah memasukkan kode untuk mengambil data input, maka tahap berikutnya adalah melakukan validasi input tadi dengan kode berikut:

Potongan Kode

```

if(NIM.equals("")||Nama.equals("")||Email.equals("")||Password.equals(""))
{
    Toast.makeText(this,"Input Data Masih Kosong",Toast.LENGTH_SHORT).show()
}
else
{
}

```

```

if(NIM.equals("")||Nama.equals("")||Email.equals("")||Password.equals(""))
{
    Toast.makeText( context: this, text: "Input Data Masih Kosong", Toast.LENGTH_SHORT).show()
}
else
{
}

```

10. Sesudah memberikan logika pengecekan input kosong, berikutnya adalah melakukan validasi E-Mail.

11. Tambahkan **import** untuk pengecekan E-Mail

Potongan Kode

```
import android.util.Patterns
```

The screenshot shows the Java code for the `DaftarMahasiswa` activity. The code includes imports for `AppCompatActivity`, `EditText`, `Toast`, and `MessageDigest`. It defines a constructor that initializes an `EditText` and sets its text to "Masukkan NIM". It also contains a method to validate an input string and another to calculate an MD5 hash.

```
package com.example.praktikum1mp

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import java.security.MessageDigest
import android.util.Patterns

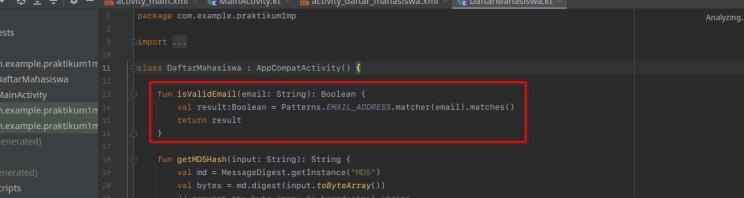
class DaftarMahasiswa : AppCompatActivity() {

    fun isValidEmail(email: String): Boolean {
        val result: Boolean =
    }

    fun getMD5Hash(input: String): String {
        val md = MessageDigest.getInstance("MD5")
        val bytes = md.digest(input.toByteArray())
        // convert the byte array to hexadecimal string
        val hexString = StringBuilder()
        for (i in bytes.indices) {
            val hex = Integer.toHexString((0xFF and bytes[i].toInt()))
            if (hex.length == 1) {
                hexString.append("0$hex")
            } else {
                hexString.append(hex)
            }
        }
        hexString.append(hex)
    }
}
```

12. Lalu tambahkan kode fungsi pengecekan Email berikut tepat di atas **override function onCreate** perhatikan gambar untuk membantu

```
Potongan Kode
fun isValidEmail(email:String):Boolean {
    val result:Boolean = Patterns.EMAIL_ADDRESS.matcher(email).matches()
    return result
}
```



```
    package com.example.praktikum1mp

    import android.os.Bundle
    import androidx.appcompat.app.AppCompatActivity
    import android.util.Patterns
    import android.widget.Toast

    class DaftarMahasiswa : AppCompatActivity() {

        fun isValidEmail(email: String): Boolean {
            val result: Boolean = Patterns.EMAIL_ADDRESS.matcher(email).matches()
            return result
        }

        fun getMD5Hash(input: String): String {
            val md = MessageDigest.getInstance("MD5")
            val bytes = md.digest(input.toByteArray())
            // convert the byte array to hexadecimal string
            val hexString = StringBuilder()
            for (i in bytes.indices) {
                val hex = Integer.toHexString(0xFF and bytes[i].toInt())
                if (hex.length == 1) {
                    hexString.append('0')
                }
                hexString.append(hex)
            }
            return hexString.toString()
        }
    }
```

13. Kembali ke kode **ELSE** yang dibuat di langkah 9, dan masukkan kode berikut untuk mengecek validitas E-Mail

Potongan Kode

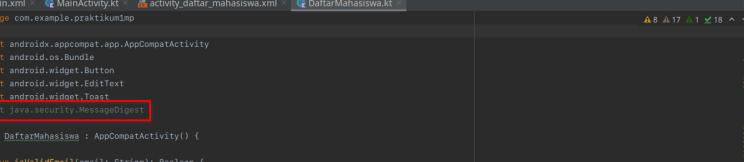
```
if(!isValidEmail(Email))
{
    Toast.makeText(this,"E-Mail Tidak Valid",Toast.LENGTH_SHORT).show()
}
else
{
}
```

```
34     val NameString = inputNameDataName.text.toString()
35     val EmailString = inputDataEmail.text.toString()
36     val PasswordString = inputDataPassword.text.toString()
37
38     if(NIM.equals("")||Name.equals("")||Email.equals("")||Password.equals(""))
39     {
40         Toast.makeText(@context this, text: "Input Data Hasil Kosong",Toast.LENGTH_SHORT).show()
41     }
42     else
43     {
44         if(!isValidEmail(Email))
45         {
46             Toast.makeText(context this, text: "E-Mail Tidak Valid",Toast.LENGTH_SHORT).show()
47         }
48         else
49         {
50
51         }
52     }
53 }
54
55 btnAskiBersih.setOnClickListener { @View
56     // Bersihkan input data
57     inputDataNIM.setText("")
58     inputDataName.setText("")
59     inputDataEmail.setText("")
```

14. Ketika kode untuk verifikasi E-Mail sudah ditambahkan, maka langkah berikutnya adalah melakukan penyamaran **Password** untuk keamanan. Tambahkan kode import berikut di bagian atas

Potongan Kode

```
import java.security.MessageDigest
```



```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
PraktikumIMP app src main java com.example.praktikumimp DafatarMahasiswa
activity_main.xml R MainActivity.kt activity_daftar_mahasiswa.xml DafatarMahasiswa.kt
activity_main.xml R MainActivity.kt activity_daftar_mahasiswa.xml DafatarMahasiswa.kt

1 package com.example.praktikumimp
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import android.widget.Button
6 import android.widget.EditText
7 import android.widget.Toast
8
8 Import java.security.MessageDigest;
9
10
11 class Daftarmahasiswa : AppCompatActivity() {
12
13     fun isValidEmail(email: String): Boolean {
14         val emailRegex = Regex("^(\\w+([-+.']\\w+)*@[\\w+([-+.']\\w+)+\\.(\\w+([-+.']\\w+)*))$")
15         return emailRegex.matches(email)
16     }
17
18     override fun onCreate(savedInstanceState: Bundle?) {
19         super.onCreate(savedInstanceState)
20         setContentView(R.layout.activity_daftar_mahasiswa)
21
22         // Kode Inisialisasi
23         var inputNama:EditText = findViewById<EditText>(R.id.inputDaftarName)
24         var inputNamaItem:EditText = findViewById<EditText>(R.id.inputDaftarName)
25         var inputEmail:EditText = findViewById<EditText>(R.id.inputDaftarEmail)
26         var inputPasssword:EditText = findViewById<EditText>(R.id.inputDaftarPassword)
27
28     }
29
30     fun saveData(): Boolean {
31         var nama = inputNama.text.toString()
32         var email = inputEmail.text.toString()
33         var password = inputPasssword.text.toString()
34
35         if (nama.isEmpty() || email.isEmpty() || password.isEmpty()) {
36             Toast.makeText(this, "All fields are required", Toast.LENGTH_SHORT).show()
37             return false
38         }
39
40         var messageDigest = MessageDigest.getInstance("SHA-256")
41         var byteData = password.toByteArray()
42         var digest = messageDigest.digest(byteData)
43
44         var hexString = StringBuilder()
45         for (i in digest.indices) {
46             var hex = Integer.toHexString(digest[i].toInt())
47             if (hex.length == 1) {
48                 hexString.append("0$hex")
49             } else {
50                 hexString.append(hex)
51             }
52         }
53
54         var hashedPassword = hexString.toString()
55
56         var sharedPref = getSharedPreferences("User", MODE_PRIVATE)
57         var editor = sharedPref.edit()
58         editor.putString("name", nama)
59         editor.putString("email", email)
60         editor.putString("password", hashedPassword)
61         editor.apply()
62
63         Toast.makeText(this, "Registration successful", Toast.LENGTH_SHORT).show()
64         return true
65     }
66
67     fun goBack() {
68         finish()
69     }
70
71     fun goHome() {
72         Intent(this, MainActivity::class.java).also { startActivity(it) }
73     }
74
75     fun goDaftarMahasiswa() {
76         Intent(this, Daftarmahasiswa::class.java).also { startActivity(it) }
77     }
78
79     fun goEditMahasiswa() {
80         Intent(this, EditMahasiswa::class.java).also { startActivity(it) }
81     }
82
83     fun goDeleteMahasiswa() {
84         Intent(this, DeleteMahasiswa::class.java).also { startActivity(it) }
85     }
86
87     fun goSearchMahasiswa() {
88         Intent(this, SearchMahasiswa::class.java).also { startActivity(it) }
89     }
90
91     fun goEditProfile() {
92         Intent(this, EditProfile::class.java).also { startActivity(it) }
93     }
94
95     fun goLogout() {
96         Intent(this, LoginActivity::class.java).also { startActivity(it) }
97     }
98
99     fun goAbout() {
100        Intent(this, About::class.java).also { startActivity(it) }
101    }
102
103    fun goFeedback() {
104        Intent(this, Feedback::class.java).also { startActivity(it) }
105    }
106
107    fun goHelp() {
108        Intent(this, Help::class.java).also { startActivity(it) }
109    }
110
111    fun goSetting() {
112        Intent(this, Setting::class.java).also { startActivity(it) }
113    }
114
115    fun goRateApp() {
116        Intent(this, RateApp::class.java).also { startActivity(it) }
117    }
118
119    fun goShareApp() {
120        Intent(this, ShareApp::class.java).also { startActivity(it) }
121    }
122
123    fun goRateApp() {
124        Intent(this, RateApp::class.java).also { startActivity(it) }
125    }
126
127    fun goShareApp() {
128        Intent(this, ShareApp::class.java).also { startActivity(it) }
129    }
130
131    fun goRateApp() {
132        Intent(this, RateApp::class.java).also { startActivity(it) }
133    }
134
135    fun goShareApp() {
136        Intent(this, ShareApp::class.java).also { startActivity(it) }
137    }
138
139    fun goRateApp() {
140        Intent(this, RateApp::class.java).also { startActivity(it) }
141    }
142
143    fun goShareApp() {
144        Intent(this, ShareApp::class.java).also { startActivity(it) }
145    }
146
147    fun goRateApp() {
148        Intent(this, RateApp::class.java).also { startActivity(it) }
149    }
150
151    fun goShareApp() {
152        Intent(this, ShareApp::class.java).also { startActivity(it) }
153    }
154
155    fun goRateApp() {
156        Intent(this, RateApp::class.java).also { startActivity(it) }
157    }
158
159    fun goShareApp() {
160        Intent(this, ShareApp::class.java).also { startActivity(it) }
161    }
162
163    fun goRateApp() {
164        Intent(this, RateApp::class.java).also { startActivity(it) }
165    }
166
167    fun goShareApp() {
168        Intent(this, ShareApp::class.java).also { startActivity(it) }
169    }
170
171    fun goRateApp() {
172        Intent(this, RateApp::class.java).also { startActivity(it) }
173    }
174
175    fun goShareApp() {
176        Intent(this, ShareApp::class.java).also { startActivity(it) }
177    }
178
179    fun goRateApp() {
180        Intent(this, RateApp::class.java).also { startActivity(it) }
181    }
182
183    fun goShareApp() {
184        Intent(this, ShareApp::class.java).also { startActivity(it) }
185    }
186
187    fun goRateApp() {
188        Intent(this, RateApp::class.java).also { startActivity(it) }
189    }
190
191    fun goShareApp() {
192        Intent(this, ShareApp::class.java).also { startActivity(it) }
193    }
194
195    fun goRateApp() {
196        Intent(this, RateApp::class.java).also { startActivity(it) }
197    }
198
199    fun goShareApp() {
200        Intent(this, ShareApp::class.java).also { startActivity(it) }
201    }
202
203    fun goRateApp() {
204        Intent(this, RateApp::class.java).also { startActivity(it) }
205    }
206
207    fun goShareApp() {
208        Intent(this, ShareApp::class.java).also { startActivity(it) }
209    }
210
211    fun goRateApp() {
212        Intent(this, RateApp::class.java).also { startActivity(it) }
213    }
214
215    fun goShareApp() {
216        Intent(this, ShareApp::class.java).also { startActivity(it) }
217    }
218
219    fun goRateApp() {
220        Intent(this, RateApp::class.java).also { startActivity(it) }
221    }
222
223    fun goShareApp() {
224        Intent(this, ShareApp::class.java).also { startActivity(it) }
225    }
226
227    fun goRateApp() {
228        Intent(this, RateApp::class.java).also { startActivity(it) }
229    }
230
231    fun goShareApp() {
232        Intent(this, ShareApp::class.java).also { startActivity(it) }
233    }
234
235    fun goRateApp() {
236        Intent(this, RateApp::class.java).also { startActivity(it) }
237    }
238
239    fun goShareApp() {
240        Intent(this, ShareApp::class.java).also { startActivity(it) }
241    }
242
243    fun goRateApp() {
244        Intent(this, RateApp::class.java).also { startActivity(it) }
245    }
246
247    fun goShareApp() {
248        Intent(this, ShareApp::class.java).also { startActivity(it) }
249    }
250
251    fun goRateApp() {
252        Intent(this, RateApp::class.java).also { startActivity(it) }
253    }
254
255    fun goShareApp() {
256        Intent(this, ShareApp::class.java).also { startActivity(it) }
257    }
258
259    fun goRateApp() {
260        Intent(this, RateApp::class.java).also { startActivity(it) }
261    }
262
263    fun goShareApp() {
264        Intent(this, ShareApp::class.java).also { startActivity(it) }
265    }
266
267    fun goRateApp() {
268        Intent(this, RateApp::class.java).also { startActivity(it) }
269    }
270
271    fun goShareApp() {
272        Intent(this, ShareApp::class.java).also { startActivity(it) }
273    }
274
275    fun goRateApp() {
276        Intent(this, RateApp::class.java).also { startActivity(it) }
277    }
278
279    fun goShareApp() {
280        Intent(this, ShareApp::class.java).also { startActivity(it) }
281    }
282
283    fun goRateApp() {
284        Intent(this, RateApp::class.java).also { startActivity(it) }
285    }
286
287    fun goShareApp() {
288        Intent(this, ShareApp::class.java).also { startActivity(it) }
289    }
290
291    fun goRateApp() {
292        Intent(this, RateApp::class.java).also { startActivity(it) }
293    }
294
295    fun goShareApp() {
296        Intent(this, ShareApp::class.java).also { startActivity(it) }
297    }
298
299    fun goRateApp() {
300        Intent(this, RateApp::class.java).also { startActivity(it) }
301    }
302
303    fun goShareApp() {
304        Intent(this, ShareApp::class.java).also { startActivity(it) }
305    }
306
307    fun goRateApp() {
308        Intent(this, RateApp::class.java).also { startActivity(it) }
309    }
310
311    fun goShareApp() {
312        Intent(this, ShareApp::class.java).also { startActivity(it) }
313    }
314
315    fun goRateApp() {
316        Intent(this, RateApp::class.java).also { startActivity(it) }
317    }
318
319    fun goShareApp() {
320        Intent(this, ShareApp::class.java).also { startActivity(it) }
321    }
322
323    fun goRateApp() {
324        Intent(this, RateApp::class.java).also { startActivity(it) }
325    }
326
327    fun goShareApp() {
328        Intent(this, ShareApp::class.java).also { startActivity(it) }
329    }
330
331    fun goRateApp() {
332        Intent(this, RateApp::class.java).also { startActivity(it) }
333    }
334
335    fun goShareApp() {
336        Intent(this, ShareApp::class.java).also { startActivity(it) }
337    }
338
339    fun goRateApp() {
340        Intent(this, RateApp::class.java).also { startActivity(it) }
341    }
342
343    fun goShareApp() {
344        Intent(this, ShareApp::class.java).also { startActivity(it) }
345    }
346
347    fun goRateApp() {
348        Intent(this, RateApp::class.java).also { startActivity(it) }
349    }
350
351    fun goShareApp() {
352        Intent(this, ShareApp::class.java).also { startActivity(it) }
353    }
354
355    fun goRateApp() {
356        Intent(this, RateApp::class.java).also { startActivity(it) }
357    }
358
359    fun goShareApp() {
360        Intent(this, ShareApp::class.java).also { startActivity(it) }
361    }
362
363    fun goRateApp() {
364        Intent(this, RateApp::class.java).also { startActivity(it) }
365    }
366
367    fun goShareApp() {
368        Intent(this, ShareApp::class.java).also { startActivity(it) }
369    }
370
371    fun goRateApp() {
372        Intent(this, RateApp::class.java).also { startActivity(it) }
373    }
374
375    fun goShareApp() {
376        Intent(this, ShareApp::class.java).also { startActivity(it) }
377    }
378
379    fun goRateApp() {
380        Intent(this, RateApp::class.java).also { startActivity(it) }
381    }
382
383    fun goShareApp() {
384        Intent(this, ShareApp::class.java).also { startActivity(it) }
385    }
386
387    fun goRateApp() {
388        Intent(this, RateApp::class.java).also { startActivity(it) }
389    }
390
391    fun goShareApp() {
392        Intent(this, ShareApp::class.java).also { startActivity(it) }
393    }
394
395    fun goRateApp() {
396        Intent(this, RateApp::class.java).also { startActivity(it) }
397    }
398
399    fun goShareApp() {
400        Intent(this, ShareApp::class.java).also { startActivity(it) }
401    }
402
403    fun goRateApp() {
404        Intent(this, RateApp::class.java).also { startActivity(it) }
405    }
406
407    fun goShareApp() {
408        Intent(this, ShareApp::class.java).also { startActivity(it) }
409    }
410
411    fun goRateApp() {
412        Intent(this, RateApp::class.java).also { startActivity(it) }
413    }
414
415    fun goShareApp() {
416        Intent(this, ShareApp::class.java).also { startActivity(it) }
417    }
418
419    fun goRateApp() {
420        Intent(this, RateApp::class.java).also { startActivity(it) }
421    }
422
423    fun goShareApp() {
424        Intent(this, ShareApp::class.java).also { startActivity(it) }
425    }
426
427    fun goRateApp() {
428        Intent(this, RateApp::class.java).also { startActivity(it) }
429    }
430
431    fun goShareApp() {
432        Intent(this, ShareApp::class.java).also { startActivity(it) }
433    }
434
435    fun goRateApp() {
436        Intent(this, RateApp::class.java).also { startActivity(it) }
437    }
438
439    fun goShareApp() {
440        Intent(this, ShareApp::class.java).also { startActivity(it) }
441    }
442
443    fun goRateApp() {
444        Intent(this, RateApp::class.java).also { startActivity(it) }
445    }
446
447    fun goShareApp() {
448        Intent(this, ShareApp::class.java).also { startActivity(it) }
449    }
450
451    fun goRateApp() {
452        Intent(this, RateApp::class.java).also { startActivity(it) }
453    }
454
455    fun goShareApp() {
456        Intent(this, ShareApp::class.java).also { startActivity(it) }
457    }
458
459    fun goRateApp() {
460        Intent(this, RateApp::class.java).also { startActivity(it) }
461    }
462
463    fun goShareApp() {
464        Intent(this, ShareApp::class.java).also { startActivity(it) }
465    }
466
467    fun goRateApp() {
468        Intent(this, RateApp::class.java).also { startActivity(it) }
469    }
470
471    fun goShareApp() {
472        Intent(this, ShareApp::class.java).also { startActivity(it) }
473    }
474
475    fun goRateApp() {
476        Intent(this, RateApp::class.java).also { startActivity(it) }
477    }
478
479    fun goShareApp() {
480        Intent(this, ShareApp::class.java).also { startActivity(it) }
481    }
482
483    fun goRateApp() {
484        Intent(this, RateApp::class.java).also { startActivity(it) }
485    }
486
487    fun goShareApp() {
488        Intent(this, ShareApp::class.java).also { startActivity(it) }
489    }
490
491    fun goRateApp() {
492        Intent(this, RateApp::class.java).also { startActivity(it) }
493    }
494
495    fun goShareApp() {
496        Intent(this, ShareApp::class.java).also { startActivity(it) }
497    }
498
499    fun goRateApp() {
500        Intent(this, RateApp::class.java).also { startActivity(it) }
501    }
502
503    fun goShareApp() {
504        Intent(this, ShareApp::class.java).also { startActivity(it) }
505    }
506
507    fun goRateApp() {
508        Intent(this, RateApp::class.java).also { startActivity(it) }
509    }
510
511    fun goShareApp() {
512        Intent(this, ShareApp::class.java).also { startActivity(it) }
513    }
514
515    fun goRateApp() {
516        Intent(this, RateApp::class.java).also { startActivity(it) }
517    }
518
519    fun goShareApp() {
520        Intent(this, ShareApp::class.java).also { startActivity(it) }
521    }
522
523    fun goRateApp() {
524        Intent(this, RateApp::class.java).also { startActivity(it) }
525    }
526
527    fun goShareApp() {
528        Intent(this, ShareApp::class.java).also { startActivity(it) }
529    }
530
531    fun goRateApp() {
532        Intent(this, RateApp::class.java).also { startActivity(it) }
533    }
534
535    fun goShareApp() {
536        Intent(this, ShareApp::class.java).also { startActivity(it) }
537    }
538
539    fun goRateApp() {
540        Intent(this, RateApp::class.java).also { startActivity(it) }
541    }
542
543    fun goShareApp() {
544        Intent(this, ShareApp::class.java).also { startActivity(it) }
545    }
546
547    fun goRateApp() {
548        Intent(this, RateApp::class.java).also { startActivity(it) }
549    }
550
551    fun goShareApp() {
552        Intent(this, ShareApp::class.java).also { startActivity(it) }
553    }
554
555    fun goRateApp() {
556        Intent(this, RateApp::class.java).also { startActivity(it) }
557    }
558
559    fun goShareApp() {
556
```

15. Setelah itu tambahkan kode untuk melakukan **hashing**. Letakkan di bawah kode pengecekan Email

Potongan Kode

```
fun getMD5Hash(input: String): String {
    val md = MessageDigest.getInstance("MD5")
    val bytes = md.digest(input.toByteArray())
    // Konversi ke Hexa
    val hexString = StringBuilder()
    for (i in bytes.indices) {
        val hex = Integer.toHexString(0xFF and bytes[i].toInt())
        if (hex.length == 1) {
            hexString.append('0')
        }
        hexString.append(hex)
    }
    return hexString.toString()
}
```

```
activity_main.xml  MainActivity.kt  activity_datar_mahasiswa.xml  DaftarMahasiswa.kt

9
10 class DaftarMahasiswa : AppCompatActivity() {
11
12     fun isValidEmail(email: String): Boolean {
13         val emailRegex = Regex("^(\\w+([-+.\\w]*\\w+))@((\\w+([-+.\\w]*\\w+))\\.(\\w+([-+.\\w]*\\w+)))$")
14         return emailRegex.matches(email)
15     }
16
17
18     fun getMD5Hash(input: String): String {
19         val md = MessageDigest.getInstance(algorithm: "MD5")
20         val bytes = md.digest(input.toByteArray())
21         // convert the byte array to hexadecimal string
22         val hexString = StringBuilder()
23         for (i in bytes.indices) {
24             val hex = Integer.toHexString(0xFF and bytes[i].toInt())
25             if (hex.length == 1) {
26                 hexString.append("0")
27             }
28             hexString.append(hex)
29         }
30         return hexString.toString()
31     }
32
33     override fun onCreate(savedInstanceState: Bundle?) {
34         super.onCreate(savedInstanceState)
35     }
36
37     override fun onDestroy() {
38         super.onDestroy()
39     }
40
41     override fun onOptionsItemSelected(item: MenuItem): Boolean {
42         when (item.itemId) {
43             R.id.action_settings -> {
44                 val intent = Intent(this, SettingsActivity::class.java)
45                 startActivity(intent)
46             }
47             else -> return super.onOptionsItemSelected(item)
48         }
49         return true
50     }
51
52     companion object {
53         const val TAG = "DaftarMahasiswa"
54     }
55
56 }
```

16. Kode ini kemudian diakses melalui kode yang ditambahkan di dalam **ELSE** dari kode langkah 11. Perhatikan kode dan gambar berikut:

Potongan Kode

```
// Dapatkan Hash Password  
val HashedPassword = getMD5Hash(Password)  
Toast.makeText(this,"NIM : "+NIM+  
    " Nama : "+Nama,Toast.LENGTH_SHORT).show()  
Toast.makeText(this,"Email : "+Email+  
    " Password : "+HashedPassword,Toast.LENGTH_SHORT).show()
```



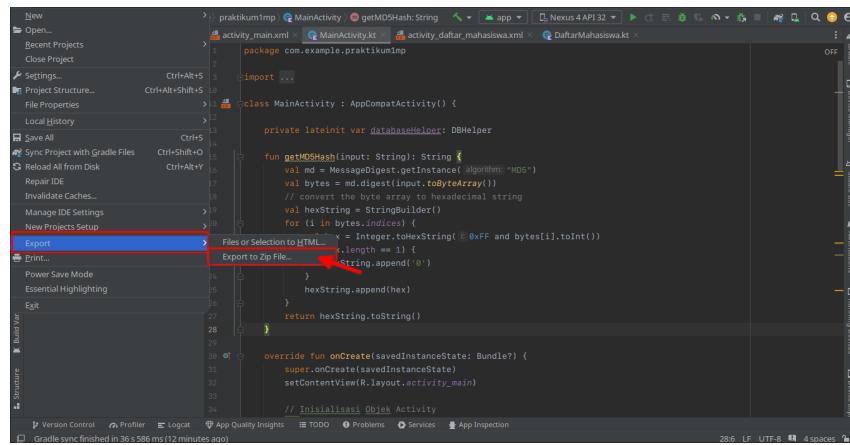
```
        {
            Toast.makeText(context, text: "Input Data Masih Kosong",Toast.LENGTH_SHORT).show()
        }
        else
        {
            if(!isValidEmail(Email))
            {
                Toast.makeText(context: this, text: "E-Mail Tidak Valid",Toast.LENGTH_SHORT).show()
            }
            else
            {
                // Bantuan Hash Password
                val HashedPassword = getMD5Hash>Password)
                Toast.makeText(context: this, text: "NIM : "+NIM+
                    " Name : "+Nama,Toast.LENGTH_SHORT).show()
                Toast.makeText(context: this, text: "Email : "+Email+
                    " Password : "+HashedPassword,Toast.LENGTH_SHORT).show()
            }
        }
    }
    btnAksiBersih.setOnClickListener { @View
        // Pembersih Layar
        inputTextNIM.setText("")
        inputTextNama.setText("")
        inputTextEmail.setText("")
        inputTextPassword.setText("")
```

17. Uji coba aplikasi dengan emmasukkan data di bagian Pendaftaran

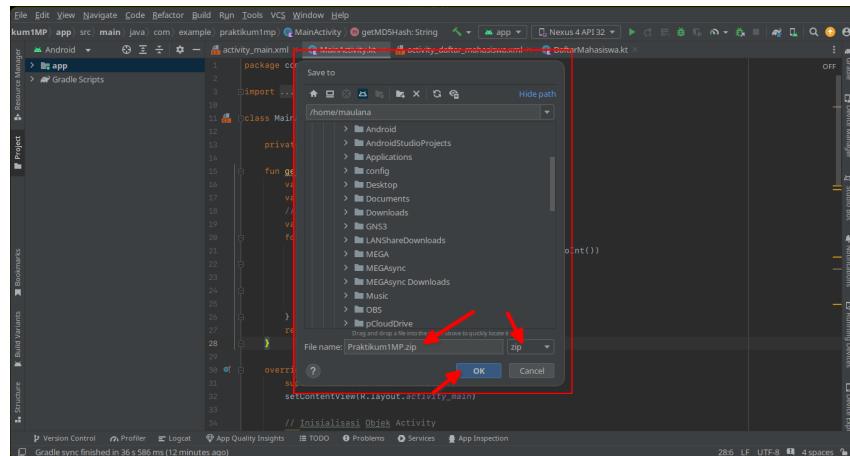
The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The project is named "Prakikulum1MP".
- Main Activity XML:** The layout file contains a single button labeled "DAFTAR".
- Main Activity Java Code:** The code handles button click events. It checks if the email is valid (using a placeholder). If invalid, it shows a toast message. If valid, it generates a hashed password and shows another toast message.
- Daftar Mahasiswa XML:** The layout file contains three input fields: "Nama", "Email", and "Password", and a "BERSIH" button.
- Daftar Mahasiswa Java Code:** The code handles the "BERSIH" button click, clearing all input fields.
- Preview:** The preview window shows the "DAFTAR" button. Below it, a modal dialog is displayed with the title "Pendaftara Data Mahasiswa" containing fields for "A11" (set to "Budi"), "Email" (set to "budi@gmail.com"), and a "DAFTAR" button.

18. Untuk mengirimkan hasilnya, klik **File**, pilih **Export**, dan pilih **Export to ZIP**.



19. Pilih lokasi yang mudah diingat, pastikan nama dan format ZIP sudah sesuai, dan klik **OK**



Bab 5

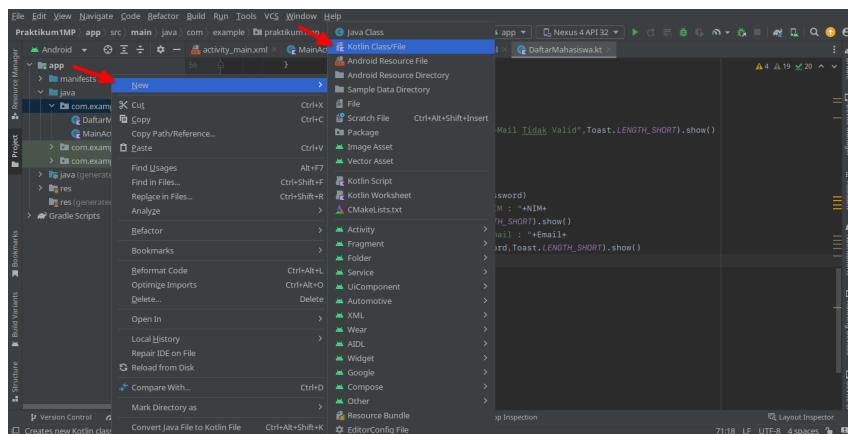
Praktikum 5

5.1 SQLite Insert Data

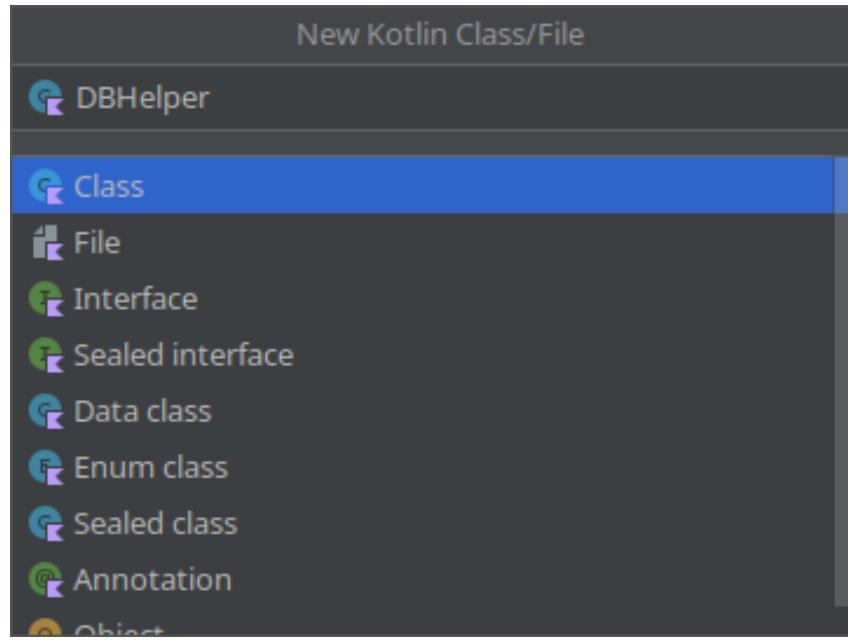
Di bagian ini mahasiswa diajarkan bagaimana melakukan input data melalui Form yang telah dibuat dan mengirimkannya ke database internal SQLite. Mahasiswa diwajibkan menyelesaikan Praktikum 4.

5.2 Tutorial

1. Praktikum dimulai dengan membuka projek sebelumnya di Android Studio.
2. Sebelum memulai, buatlah satu file **Kotlin** untuk membantu mengatur akses database SQLite. Klik kanan folder **com.example.Praktikum1MP**, pilih **New**, pilih **Kotlin Class/File**



3. Masukkan nama file **DBHelper** lalu tekan **Enter**



4. Masukkan kode berikut untuk file DBHelper. Jangan hapus baris kode package karena sangat vital.

Potongan Kode

```
import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper

class DBHelper(context: Context) : SQLiteOpenHelper(context,
    DATABASE_NAME, null, DATABASE_VERSION) {

    companion object {
        private const val DATABASE_NAME = "DB_MHS"
        private const val DATABASE_VERSION = 1

        private const val CREATE_TABLE_MY_DATA = "CREATE TABLE TBL_MHS " +
            "(nim TEXT PRIMARY KEY, nama TEXT, email TEXT," +
            "password TEXT)"
    }

    override fun onCreate(db: SQLiteDatabase) {
        db.execSQL(CREATE_TABLE_MY_DATA)
    }

    override fun onUpgrade(db: SQLiteDatabase, oldVersion: Int,
        newVersion: Int) {
        // Untuk upgrade database
    }
}
```

```

1 package com.example.praktikumImp;
2
3 import android.content.Context;
4 import android.database.sqlite.SQLiteDatabase;
5 import android.database.sqlite.SQLiteOpenHelper;
6
7
8 class DBHelper extends SQLiteOpenHelper {
9
10     private const val DATABASE_NAME = "DB_JHS";
11     private const val DATABASE_VERSION = 1;
12
13     // Definisi Tabel
14     private const val CREATE_TABLE_MY_DATA = "CREATE TABLE TBL_MHS" +
15             "(nim TEXT PRIMARY KEY, name TEXT, email TEXT," +
16             "password TEXT)";
17
18
19     @Override
20     public void onCreate(SQLiteDatabase db) {
21         db.execSQL(CREATE_TABLE_MY_DATA);
22     }
23
24     @Override
25     public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
26         // Untuk upgrade database
27     }
28 }

```

5. Buka file DaftarMahasiswa.kt

```

1 package com.example.praktikumImp;
2
3 import android.os.Bundle;
4 import android.util.Patterns;
5 import android.view.View;
6 import android.widget.EditText;
7 import android.widget.Toast;
8
9 import androidx.appcompat.app.AppCompatActivity;
10
11 public class DaftarMahasiswa extends AppCompatActivity {
12
13     EditText inputDataNIM, inputDataNama, inputDataEmail, inputDataPassword;
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_daftar_mahasiswa);
19
20         inputDataNIM = findViewById(R.id.inputDataNIM);
21         inputDataNama = findViewById(R.id.inputDataNama);
22         inputDataEmail = findViewById(R.id.inputDataEmail);
23         inputDataPassword = findViewById(R.id.inputDataPassword);
24
25         btnAksiBersih.setOnClickListener(new View.OnClickListener() {
26             @Override
27             public void onClick(View v) {
28                 // Bersihkan input data
29                 inputDataNIM.setText("");
30                 inputDataNama.setText("");
31                 inputDataEmail.setText("");
32                 inputDataPassword.setText("");
33             }
34         });
35         btnAksiBatal.setOnClickListener(new View.OnClickListener() {
36             @Override
37             public void onClick(View v) {
38                 // Batalkan dengan finish
39             }
40         });
41     }
42
43     private void checkInput() {
44         String NIM = inputDataNIM.getText().toString();
45         String Name = inputDataNama.getText().toString();
46         String Email = inputDataEmail.getText().toString();
47         String Password = inputDataPassword.getText().toString();
48
49         if (!isValidEmail(Email)) {
50             Toast.makeText(context, "Email Tidak Valid", Toast.LENGTH_SHORT).show();
51         } else {
52             // Dapatkan Hash Password
53             String HashedPassword = getMD5Hash>Password);
54             Toast.makeText(context, "NIM : " + NIM +
55                     "\nName : " + Name, Toast.LENGTH_SHORT).show();
56             Toast.makeText(context, "Email : " + Email +
57                     "\nPassword : " + HashedPassword, Toast.LENGTH_SHORT).show();
58         }
59     }
60
61     private boolean isValidEmail(String Email) {
62         return Patterns.EMAIL_ADDRESS.matcher(Email).matches();
63     }
64
65     private String getMD5Hash(String Password) {
66         try {
67             MessageDigest md = MessageDigest.getInstance("MD5");
68             byte[] bytes = md.digest(Password.getBytes("UTF-8"));
69             byte[] hexArray = new byte[bytes.length * 2];
70             for (int i = 0; i < bytes.length; i++) {
71                 hexArray[i * 2] = (byte) ((bytes[i] & 0xFF) >> 4);
72                 hexArray[i * 2 + 1] = (byte) (bytes[i] & 0x0F);
73             }
74             StringBuffer hexString = new StringBuffer();
75             for (byte b : hexArray) {
76                 String hex = Integer.toHexString(b & 0xFF);
77                 if (hex.length() == 1) {
78                     hexString.append('0');
79                 }
80                 hexString.append(hex);
81             }
82             return hexString.toString();
83         } catch (Exception e) {
84             e.printStackTrace();
85         }
86         return null;
87     }
88 }

```

6.Tambahkan kode untuk mengakses database, letakkan di bawah kode Class seperti contoh berikut:

Potongan Kode
private lateinit var databaseHelper: DBHelper

```

1 package com.example.praktikumImp;
2
3 import android.os.Bundle;
4 import android.util.Patterns;
5 import android.view.View;
6 import android.widget.EditText;
7 import android.widget.Toast;
8
9 import androidx.appcompat.app.AppCompatActivity;
10
11 public class DaftarMahasiswa extends AppCompatActivity {
12
13     EditText inputDataNIM, inputDataNama, inputDataEmail, inputDataPassword;
14
15     private lateinit var databaseHelper: DBHelper;
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_daftar_mahasiswa);
21
22         inputDataNIM = findViewById(R.id.inputDataNIM);
23         inputDataNama = findViewById(R.id.inputDataNama);
24         inputDataEmail = findViewById(R.id.inputDataEmail);
25         inputDataPassword = findViewById(R.id.inputDataPassword);
26
27         btnAksiBersih.setOnClickListener(new View.OnClickListener() {
28             @Override
29             public void onClick(View v) {
30                 // Bersihkan input data
31                 inputDataNIM.setText("");
32                 inputDataNama.setText("");
33                 inputDataEmail.setText("");
34                 inputDataPassword.setText("");
35             }
36         });
37         btnAksiBatal.setOnClickListener(new View.OnClickListener() {
38             @Override
39             public void onClick(View v) {
40                 // Batalkan dengan finish
41             }
42         });
43
44         checkInput();
45     }
46
47     private void checkInput() {
48         String NIM = inputDataNIM.getText().toString();
49         String Name = inputDataNama.getText().toString();
50         String Email = inputDataEmail.getText().toString();
51         String Password = inputDataPassword.getText().toString();
52
53         if (!isValidEmail(Email)) {
54             Toast.makeText(context, "Email Tidak Valid", Toast.LENGTH_SHORT).show();
55         } else {
56             // Dapatkan Hash Password
57             String HashedPassword = getMD5Hash>Password);
58             Toast.makeText(context, "NIM : " + NIM +
59                     "\nName : " + Name, Toast.LENGTH_SHORT).show();
60             Toast.makeText(context, "Email : " + Email +
61                     "\nPassword : " + HashedPassword, Toast.LENGTH_SHORT).show();
62         }
63     }
64
65     private boolean isValidEmail(String Email) {
66         return Patterns.EMAIL_ADDRESS.matcher(Email).matches();
67     }
68
69     private String getMD5Hash(String Password) {
70         try {
71             MessageDigest md = MessageDigest.getInstance("MD5");
72             byte[] bytes = md.digest(Password.getBytes("UTF-8"));
73             byte[] hexArray = new byte[bytes.length * 2];
74             for (int i = 0; i < bytes.length; i++) {
75                 hexArray[i * 2] = (byte) ((bytes[i] & 0xFF) >> 4);
76                 hexArray[i * 2 + 1] = (byte) (bytes[i] & 0x0F);
77             }
78             StringBuffer hexString = new StringBuffer();
79             for (byte b : hexArray) {
80                 String hex = Integer.toHexString(b & 0xFF);
81                 if (hex.length() == 1) {
82                     hexString.append('0');
83                 }
84                 hexString.append(hex);
85             }
86             return hexString.toString();
87         } catch (Exception e) {
88             e.printStackTrace();
89         }
90         return null;
91     }
92 }

```

7. Tambahkan lagi inisialisasi lokal tepat di bawah kode inisialisasi Widget Layout. Perhatikan gambar dan kode berikut:

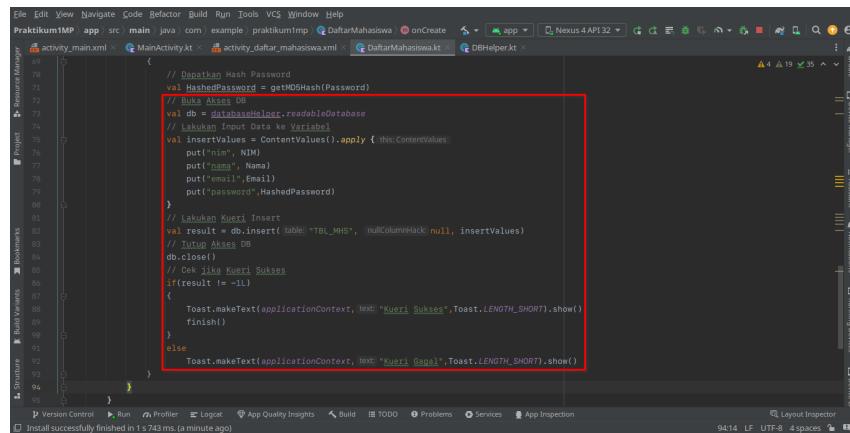
Potongan Kode
databaseHelper = DBHelper(this)

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
PraktikumIMP app src main java com.example.praktikumimp DaftarMahasiswa onCreate DBHelper.kt
activity_main.xml MainActivity.kt activity_daftar_mahasiswa.xml DBHelper.kt

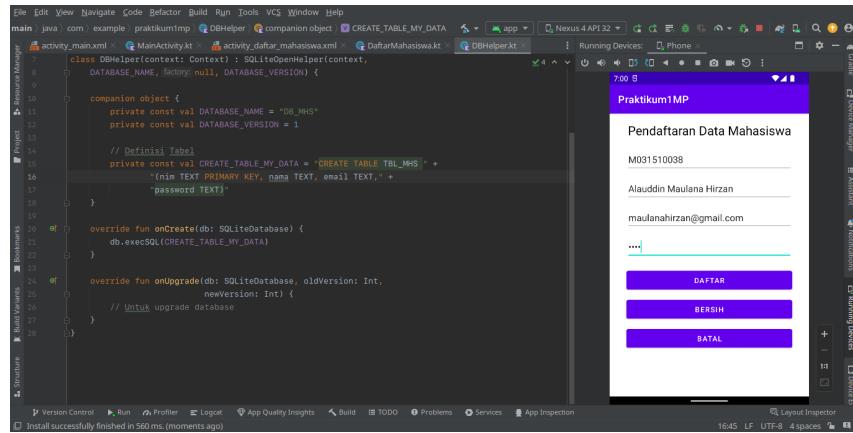
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2278
2
```

Potongan Kode

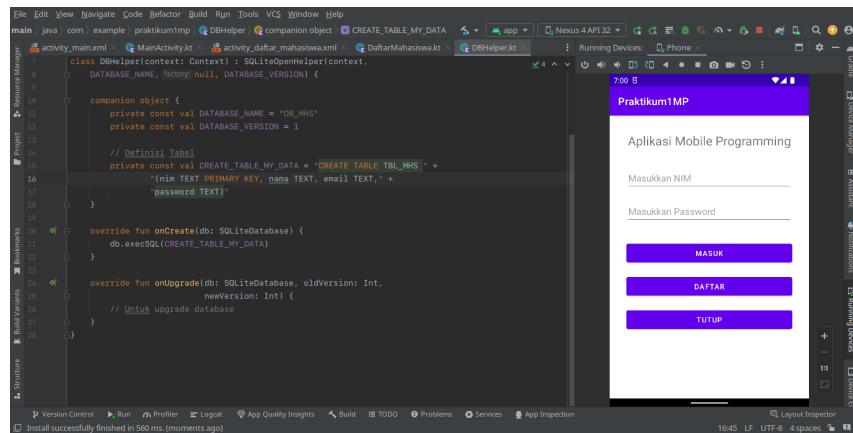
```
// Buka Akses DB
val db = databaseHelper.readableDatabase
// Lakukan Input Data ke Variabel
val insertValues = ContentValues().apply {
    put("nim", NIM)
    put("nama", Nama)
    put("email", Email)
    put("password", HashedPassword)
}
// Lakukan Kueri Insert
val result = db.insert("TBL_MHS", null, insertValues)
// Tutup Akses DB
db.close()
// Cek jika Kueri Sukses
if(result != -1L)
{
    Toast.makeText(applicationContext, "Kueri Sukses", Toast.LENGTH_SHORT).show()
    finish()
}
else
    Toast.makeText(applicationContext, "Kueri Gagal", Toast.LENGTH_SHORT).show()
```



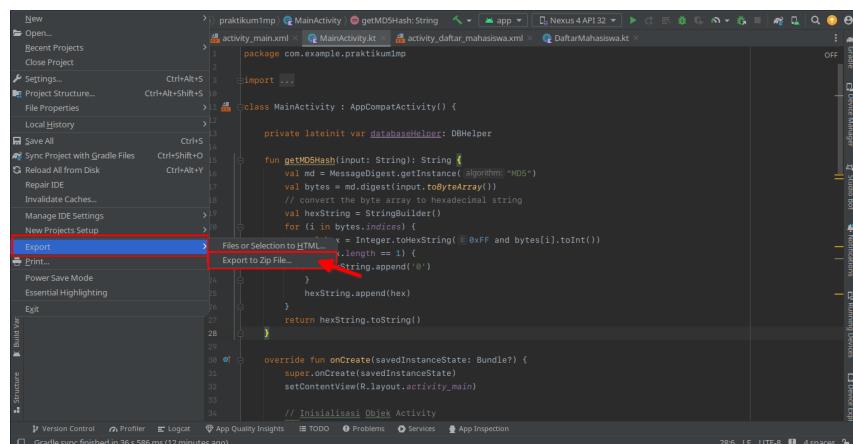
10. Lakukan import untuk **ContentValues** jika terjadi error
11. Jalankan **Emulator** terlebih dahulu, lalu buka jalankan aplikasi. Coba kirimkan data melalui form pendaftaran



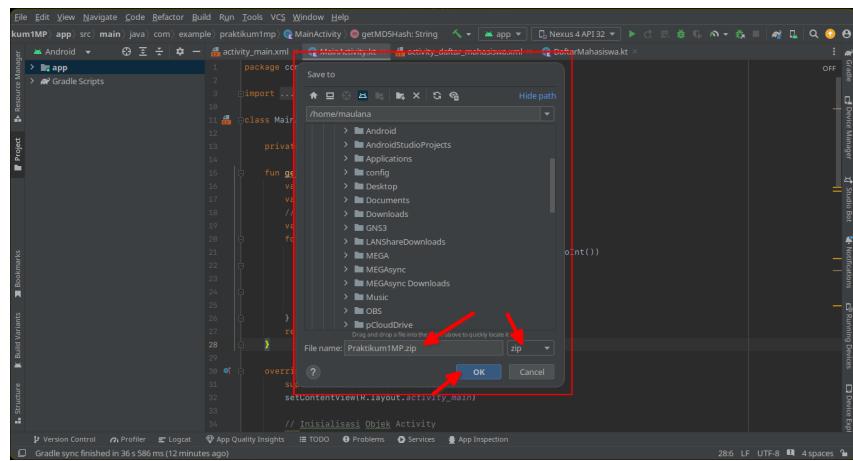
12. Jika kueri berhasil maka akan muncul **Toast** dan halaman akan kembali ke halaman **Login**



13. Untuk mengirimkan hasilnya, klik **File**, pilih **Export**, dan pilih **Export to ZIP**.



14. Pilih lokasi yang mudah diingat, pastikan nama dan format ZIP sudah sesuai, dan klik **OK**



Bab 6

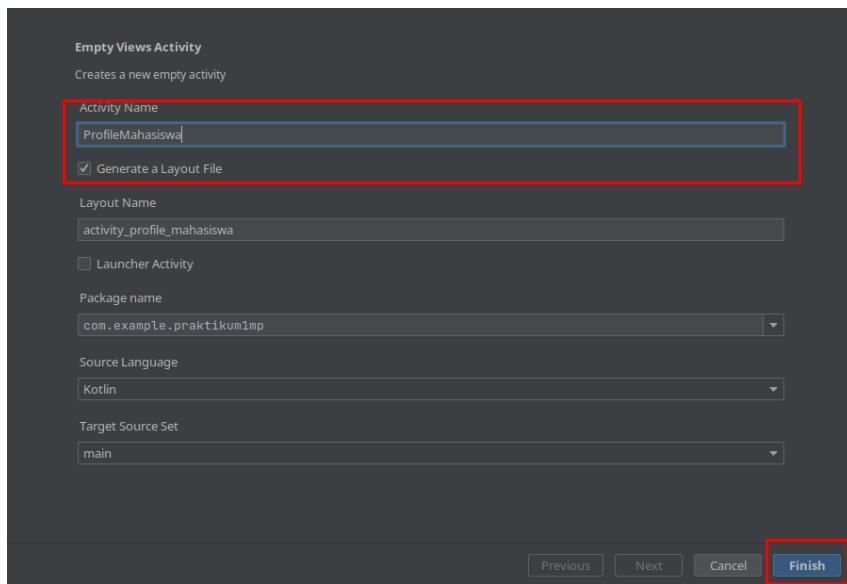
Praktikum 6

6.1 SQLite Read Data

Di bagian ini mahasiswa diajarkan bagaimana melakukan pengambilan data untuk login maupun untuk halaman profil. Mahasiswa diwajibkan untuk menyelesaikan Praktikum 5 sebelum melakukan praktikum ini

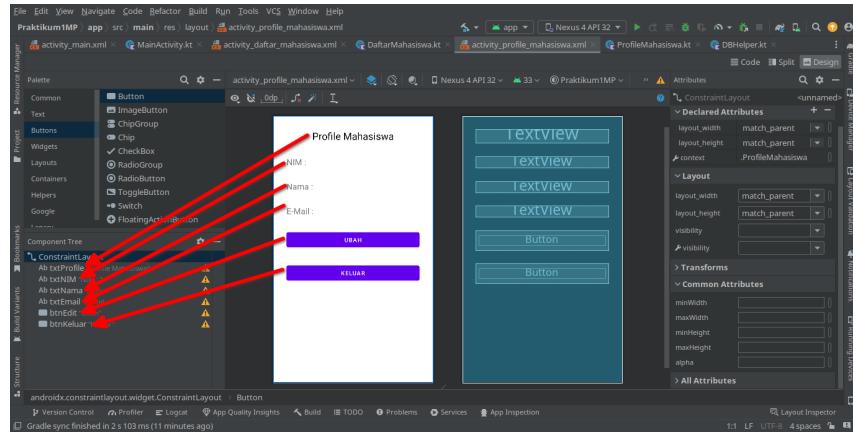
6.2 Tutorial

1. Buka kembali Android Studio dan buatlah satu **Activity** baru dengan nama **ProfileMahasiswa**

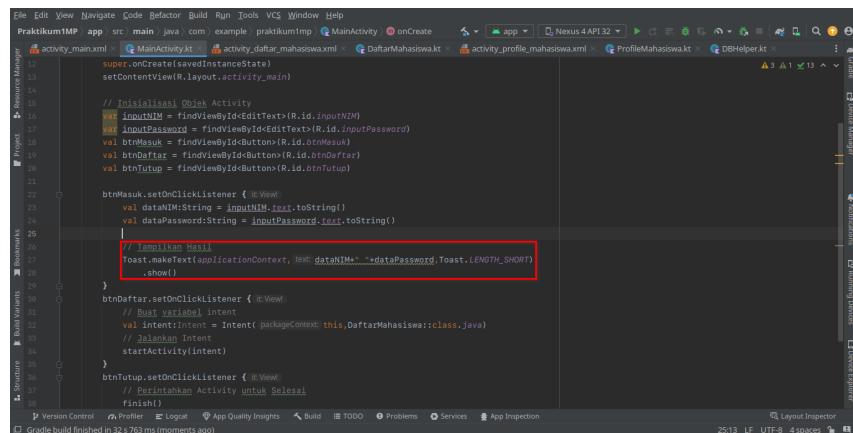


2. Buatlah tampilan halaman untuk **ProfileMahasiswa** di layout **activity_profile_mahasiswa.xml**. Perhatikan daftar objek berikut ini
 - TextView : **txtProfile** → Text : **Profile Mahasiswa**
 - TextView : **txtNIM** → Text : **NIM :**

- TextView : txtNama → Text : Nama :
- TextView : txtEmail → Text : E-Mail :

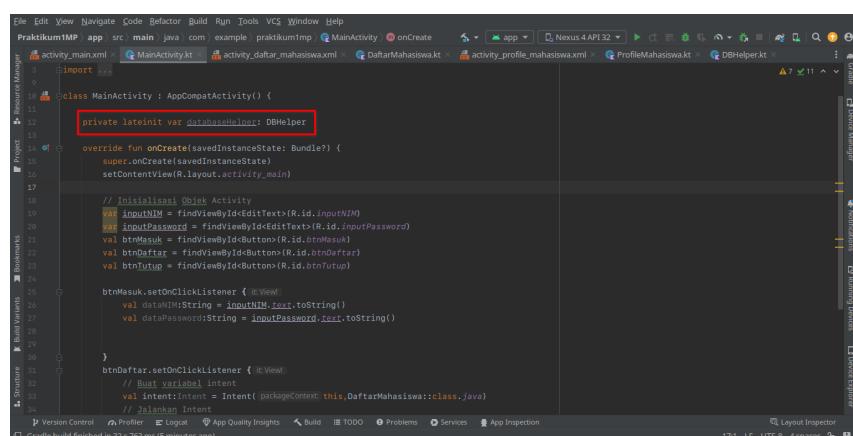


3. Sesudah layout untuk **ProfileMahasiswa** sudah jadi, berikutnya buka file **MainActivity.kt** dan **hapus** bagian kode berikut



4. Sesudah dihapus, tambahkan late init Database di bawah baris Class **MainActivity**

Potongan Kode
private lateinit var databaseHelper: DBHelper



5. Inisialisasikan database dan letakkan di bawah kode-kode inisialisasi lainnya. Lihat contoh kode dan gambar

Potongan Kode

```
databaseHelper = DBHelper(this)
```

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Praktikum1MP app src main java com example praktikum1mp MainActivity onCreate DaftarMahasiswa.kt activity_daftar_mahasiswa.xml ProfileMahasiswa.kt DBHelper.kt
13
14     override fun onCreate(savedInstanceState: Bundle?) {
15         super.onCreate(savedInstanceState)
16         setContentView(R.layout.activity_main)
17
18         // Inisialisasi Objek Activity
19         val inputNIM = findViewById<EditText>(R.id.inputNIM)
20         val inputPassword = findViewById<EditText>(R.id.inputPassword)
21         val btnMasuk = findViewById<Button>(R.id.btnMasuk)
22         val btnDaftar = findViewById<Button>(R.id.btnDaftar)
23         val btnTutup = findViewById<Button>(R.id.btnTutup)
24
25         databaseHelper = DBHelper(context: this)
26
27         btnMasuk.setOnClickListener { it:View ->
28             val dataNIM:String = inputNIM.text.toString()
29             val dataPassword:String = inputPassword.text.toString()
30
31         }
32         btnDaftar.setOnClickListener { it:View -
33             // Bust variable Intent
34             val intent:Intent = Intent(packageContext: this, DaftarMahasiswa::class.java)
35             // Jalankan Intent
36             startActivity(intent)
37         }
38         btnTutup.setOnClickListener { it:View -
39             // Perintahkan Activity untuk Selesai
40         }
41     }
42
43     override fun onSaveInstanceState(state: Bundle) {
44         super.onSaveInstanceState(state)
45     }
46
47     companion object {
48         const val TAG = "MainActivity"
49     }
50
51 }
```

6. Setelah itu tambahkan kode untuk melakukan konversi password ke checksum hash. Perhatikan kode berikut dan letakkan di bawah kode **late init**:

Potongan Kode

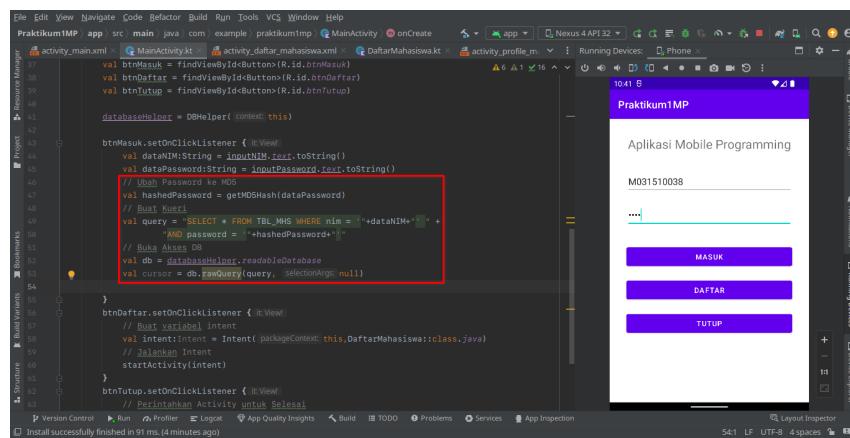
```
fun getMD5Hash(input: String): String {
    val md = MessageDigest.getInstance("MD5")
    val bytes = md.digest(input.toByteArray())
    // Konversi ke Hexa
    val hexString = StringBuilder()
    for (i in bytes.indices) {
        val hex = Integer.toHexString(0xFF and bytes[i].toInt())
        if (hex.length == 1) {
            hexString.append('0')
        }
        hexString.append(hex)
    }
    return hexString.toString()
}
```

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Praktikum1MP app src main java com example praktikum1mp MainActivity onCreate getMD5Hash DaftarMahasiswa.kt activity_daftar_mahasiswa.xml ProfileMahasiswa.kt DBHelper.kt
1
2
3 import ...
4
5 class MainActivity : AppCompatActivity() {
6
7     private lateinit var databaseHelper: DBHelper
8
9     fun getMD5Hash(input: String): String {
10         val md = MessageDigest.getInstance(algorithm: "MD5")
11         val bytes = md.digest(input.toByteArray())
12         // convert the byte array to hexadecimal string
13         val hexString = StringBuilder()
14         for (i in bytes.indices) {
15             val hex = Integer.toHexString(0xFF and bytes[i].toInt())
16             if (hex.length == 1) {
17                 hexString.append('0')
18             }
19             hexString.append(hex)
20         }
21         return hexString.toString()
22     }
23
24     override fun onCreate(savedInstanceState: Bundle?) {
25         super.onCreate(savedInstanceState)
26         setContentView(R.layout.activity_main)
27     }
28
29     companion object {
30         const val TAG = "MainActivity"
31     }
32 }
```

7. Berikutnya adalah memasukkan kode untuk verifikasi login, kembali ke bagian yang dihapus tadi (kode **btnMasuk**), dan masukkan kode berikut

Potongan Kode

```
// Ubah Password ke MD5
val hashedPassword = getMD5Hash(dataPassword)
// Buat Kueri
val query = "SELECT * FROM TBL_MHS WHERE nim = '"+dataNIM+"' " +
    "AND password = '"+hashedPassword+"'"
// Buka Akses DB
val db = databaseHelper.readableDatabase
val cursor = db.rawQuery(query, null)
```



8. Kemudian lanjutkan kode untuk pengecekan hasil kueri, perhatikan kode dan gambar berikut:

Potongan Kode

```
// Cek Hasil Kueri
val result = cursor.moveToFirst()
if(result == true)
{
    // Login Benar
}
else
{
    // Login Salah -> Bersihkan Semua
}
```

```

    val result = cursor.moveToFirst()
    if(result == true)
    {
        // Login Benar
    }
    else
    {
        // Login Salah -> Bersihkan Semua
        btnDaftar.setOnClickListener { it: View ->
            // Bust variabel intent
            val intent: Intent = Intent(packageContext: this, DaftarMahasiswa::class.java)
            // Jalankan Intent
        }
    }
}

```

9. Ketika kode **IF** sudah dimasukkan sekarang, isi bagian **Login Salah** dengan kode berikut:

Potongan Kode

```

Toast.makeText(applicationContext, "Data Login Salah",
    Toast.LENGTH_SHORT).show()
inputNIM.setText("")
inputPassword.setText("")

```

```

    val result = cursor.moveToFirst()
    if(result == true)
    {
        // Login Benar
    }
    else
    {
        // Login Salah -> Bersihkan Semua
        Toast.makeText(applicationContext, "Data Login Salah",
            Toast.LENGTH_SHORT).show()
        inputNIM.setText("")
        inputPassword.setText("")
    }
}

```

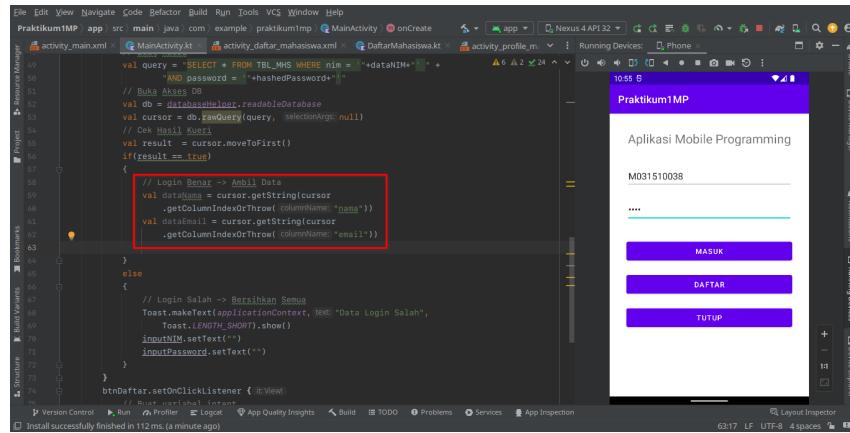
10. Lalu masukkan kode untuk mengambil data untuk **Login Benar**

Potongan Kode

```

val dataNama = cursor.getString(cursor
    .getColumnIndexOrThrow("nama"))
val dataEmail = cursor.getString(cursor
    .getColumnIndexOrThrow("email"))

```



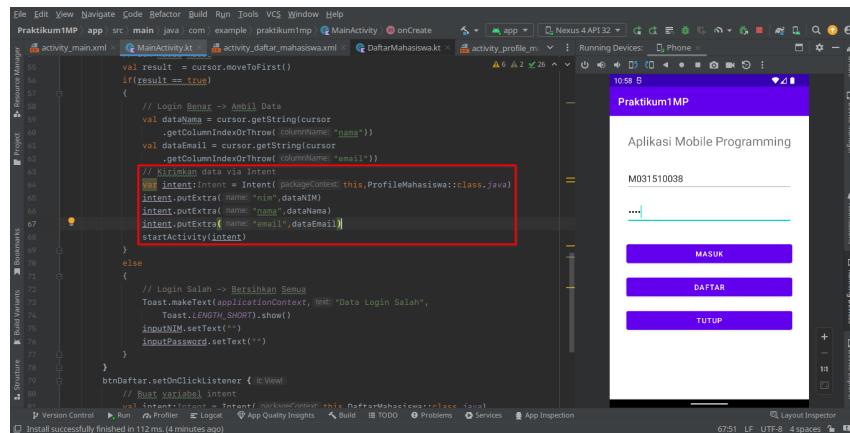
11. Setelah itu masukkan kode berikut untuk mengirimkan data-data tersebut ke halaman **ProfileMahasiswa**

Potongan Kode

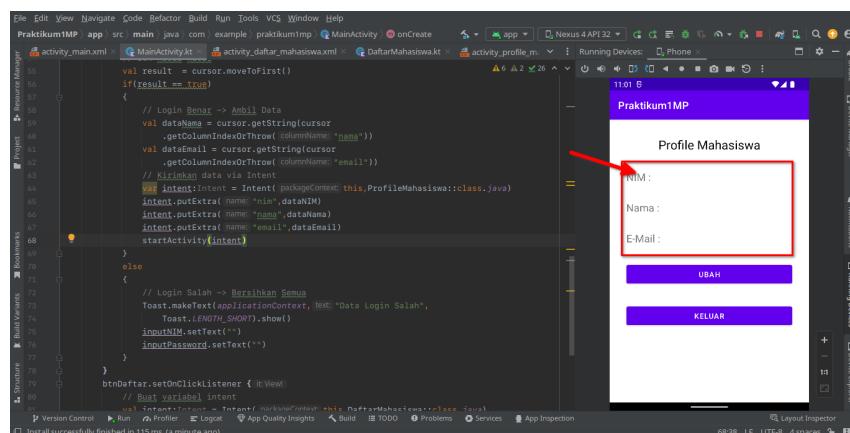
```

// Kirimkan data via Intent
var intent:Intent = Intent(this,ProfileMahasiswa::class.java)
intent.putExtra("nim",dataNIM)
intent.putExtra("nama",dataNama)
intent.putExtra("email",dataEmail)
startActivity(intent)

```



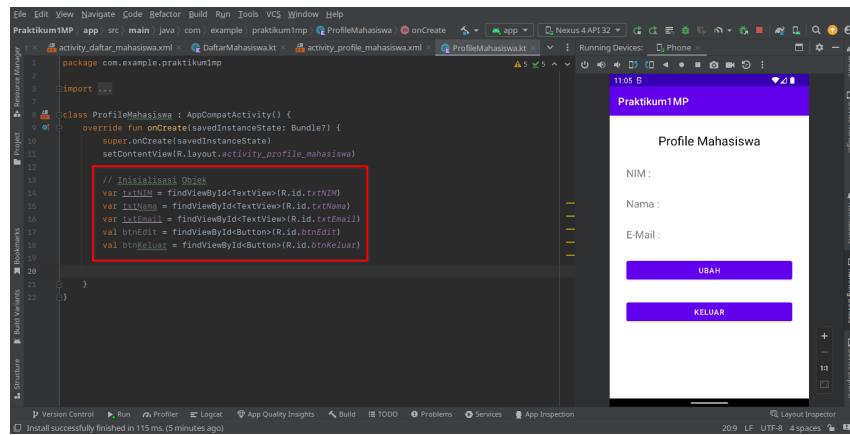
12. Jika dilakukan dengan benar, maka ketika Login aplikasi akan menampilkan Halaman Profile Mahasiswa. Tetapi masih kosong.



13. Agar data dapat ditampilkan dengan benar, masukkan kode **Inisialisasi** terlebih dahulu seperti berikut di file **ProfileMahasiswa.kt**. Perhatikan contoh Kode dan Gambar:

Potongan Kode

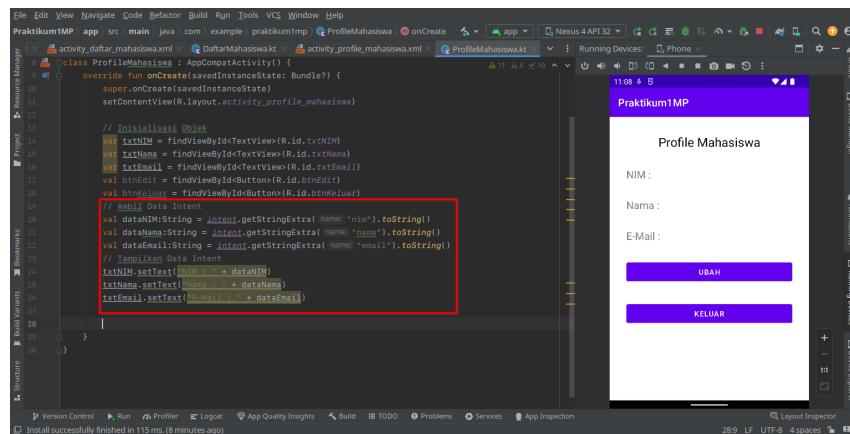
```
// Inisialisasi Objek
var txtNIM = findViewById<TextView>(R.id.txtNIM)
var txtNama = findViewById<TextView>(R.id.txtNama)
var txtEmail = findViewById<TextView>(R.id.txtEmail)
val btnEdit = findViewById<Button>(R.id.btnEdit)
val btnKeluar = findViewById<Button>(R.id.btnKeluar)
```



14. Berikutnya adalah mengambil data **Intent** dan menampilkannya ke layar. Perhatikan kode berikut:

Potongan Kode

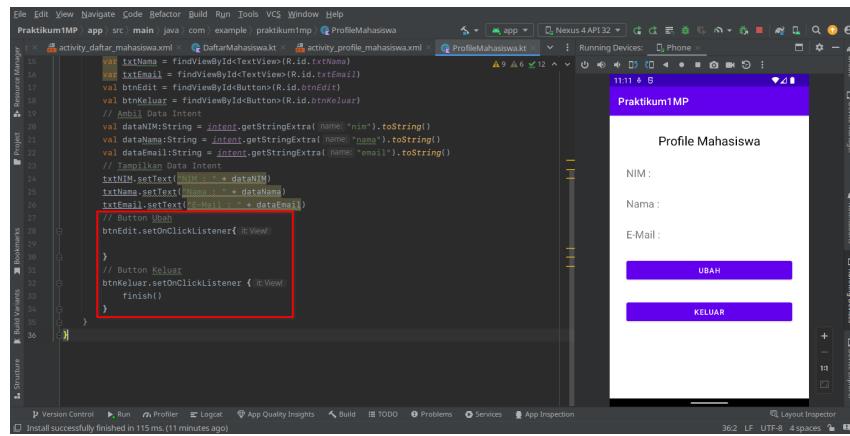
```
// Ambil Data Intent
val dataNIM:String = intent.getStringExtra("nim").toString()
val dataNama:String = intent.getStringExtra("nama").toString()
val dataEmail:String = intent.getStringExtra("email").toString()
// Tampilkan Data Intent
txtNIM.setText("NIM : " + dataNIM)
txtNama.setText("Nama : " + dataNama)
txtEmail.setText("E-Mail : " + dataEmail)
```



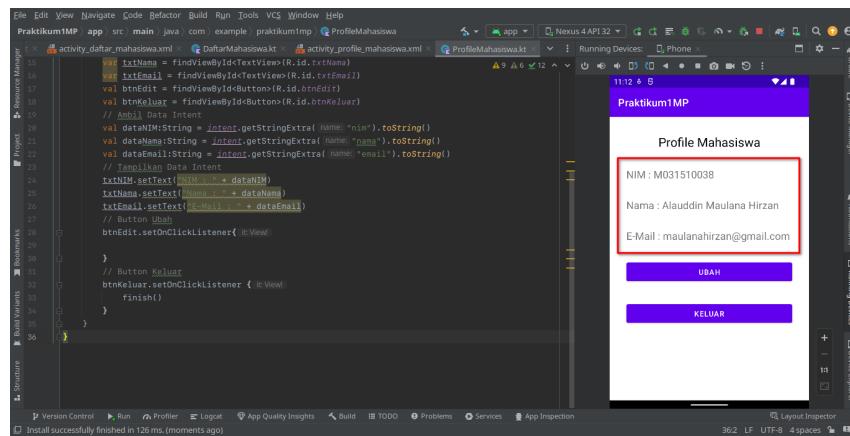
15. Terakhir tambahkan kode **Listener** untuk masing-masing **Button**

Potongan Kode

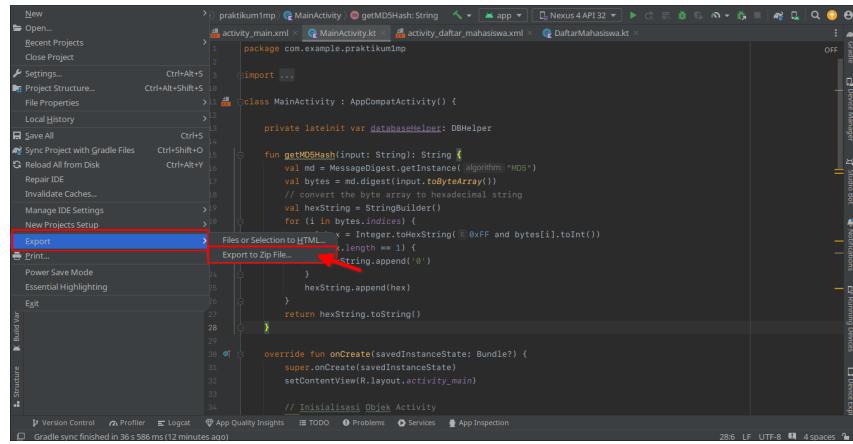
```
// Button Ubah  
btnEdit.setOnClickListener{  
  
}  
// Button Keluar  
btnKeluar.setOnClickListener {  
    finish()  
}
```



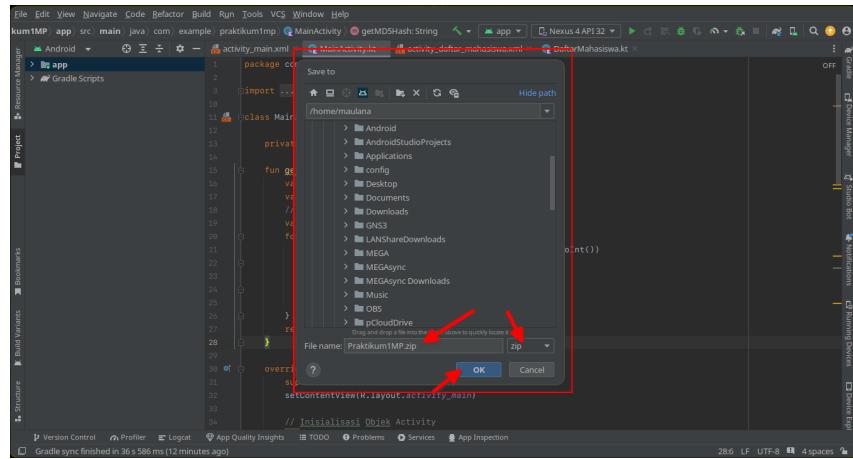
16. Nyalakan Emulator, Jalankan Aplikasi, dan Cobalah Login



17. Untuk mengirimkan hasilnya, klik File, pilih Export, dan pilih Export to ZIP.



18. Pilih lokasi yang mudah diingat, pastikan nama dan format ZIP sudah sesuai, dan klik **OK**



Bab 7

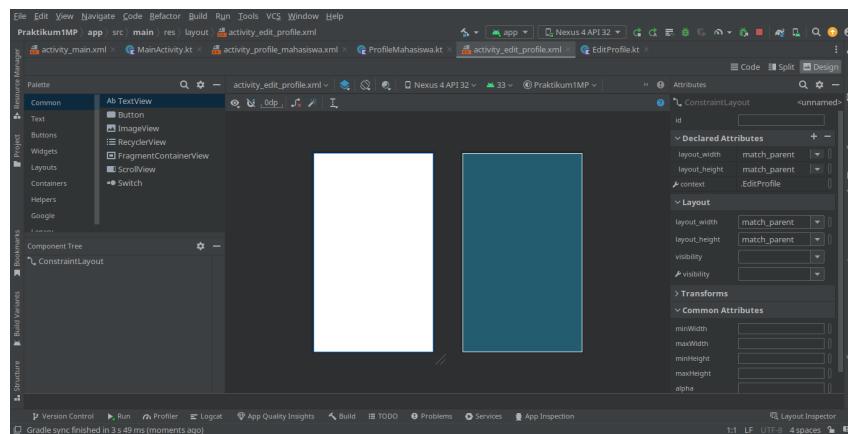
Praktikum 7

7.1 SQLite Update Data

Di bagian ini mahasiswa diajarkan bagaimana melakukan perubahan data yang ada di database. Mahasiswa diwajibkan menyelesaikan Praktikum 6 sebelum melanjutkan praktikum ini

7.2 Tutorial

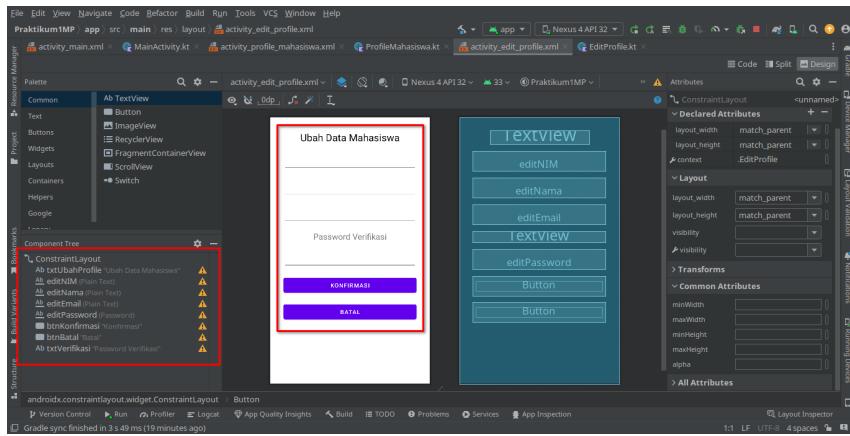
1. Buka Android Studio, dan buatlah satu **Activity Baru** dengan nama **UbahProfile**. Lalu buka layout **activity_edit_profile.xml**



2. Buatlah tampilan seperti berikut:

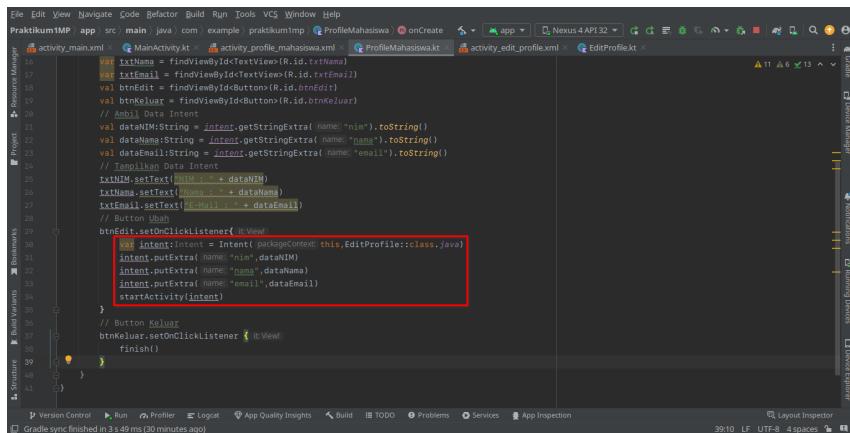
- TextView → **txtUbahProfile** : Text → **Ubah Data Mahasiswa**
- EditText Plain → **editNIM** : Text → (Kosong)
- EditText Plain → **editNama** : Text → (Kosong)
- EditText Plain → **editEmail** : Text → (Kosong)
- TextView → **txtVerifikasi** : Text → **Password Verifikasi**

- EditText Password → **editPassword** : Text → (Kosong)
- Button → **btnKonfirmasi** : Text → Konfirmasi
- Button → **btnBatal** : Text → Batal



3. Setelah selesai dengan tampilan, kembali ke **ProfileMahasiswa** dan tambahkan kode berikut untuk **btnEdit**

```
Potongan Kode
var intent:Intent = Intent(this,EditProfile::class.java)
intent.putExtra("nim",dataNIM)
intent.putExtra("nama",dataNama)
intent.putExtra("email",dataEmail)
startActivity(intent)
```



4. Berikutnya buka file **EditProfile.kt** kembali, dan tambahkan kode inisialisasi seperti contoh berikut:

```
Potongan Kode
// Kode Inisialisasi
var editNIM = findViewById<EditText>(R.id.editNIM)
var editNama = findViewById<EditText>(R.id.editNama)
var editEmail = findViewById<EditText>(R.id.editEmail)
var editPassword = findViewById<EditText>(R.id.editPassword)
```

```

1 package com.example.praktikum1mp
2
3 import ...
4
5 class EditProfile : AppCompatActivity() {
6     override fun onCreate(savedInstanceState: Bundle?) {
7         super.onCreate(savedInstanceState)
8         setContentView(R.layout.activity_edit_profile)
9
10        // Kode Inisialisasi
11        var editNIM = findViewById<EditText>(R.id.editNIM)
12        var editNama = findViewById<EditText>(R.id.editNama)
13        var editEmail = findViewById<EditText>(R.id.editEmail)
14        var editPassword = findViewById<EditText>(R.id.editPassword)
15
16    }
17
18 }
19

```

5. Tambahkan kode berikut untuk mengambil data dari Intent dan mengunci akses editNIM

Potongan Kode

```

// Ambil dan Tampilkan Data Intent
val dataNIM:String = intent.getStringExtra("nim").toString()
val dataNama:String = intent.getStringExtra("nama").toString()
val dataEmail:String = intent.getStringExtra("email").toString()
editNIM.setText(dataNIM)
editNama.setText(dataNama)
editEmail.setText(dataEmail)
// Kunci Akses editNIM
editNIM.isFocusable = false
editNIM.isFocusableInTouchMode = false

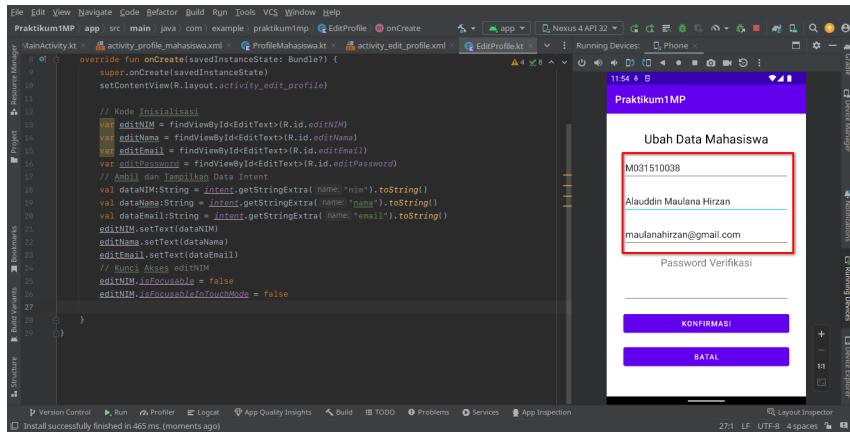
```

```

1 package com.example.praktikum1mp
2
3 import ...
4
5 class EditProfile : AppCompatActivity() {
6     override fun onCreate(savedInstanceState: Bundle?) {
7         super.onCreate(savedInstanceState)
8         setContentView(R.layout.activity_edit_profile)
9
10        // Kode Inisialisasi
11        var editNIM = findViewById<EditText>(R.id.editNIM)
12        var editNama = findViewById<EditText>(R.id.editNama)
13        var editEmail = findViewById<EditText>(R.id.editEmail)
14        var editPassword = findViewById<EditText>(R.id.editPassword)
15
16        // Ambil dan Tampilkan Data Intent
17        val dataNIMString = intent.getStringExtra("nim").toString()
18        val dataNamaString = intent.getStringExtra("nama").toString()
19        val dataEmailString = intent.getStringExtra("email").toString()
20
21        editNIM.setText(dataNIMString)
22        editNama.setText(dataNamaString)
23        editEmail.setText(dataEmailString)
24
25        // Kunci Akses editNIM
26        editNIM.isFocusable = false
27        editNIM.isFocusableInTouchMode = false
28
29    }
30

```

6. Jalankan Emulator dan Aplikasi untuk mengetes. NIM tidak akan pernah bisa dipencet / klik

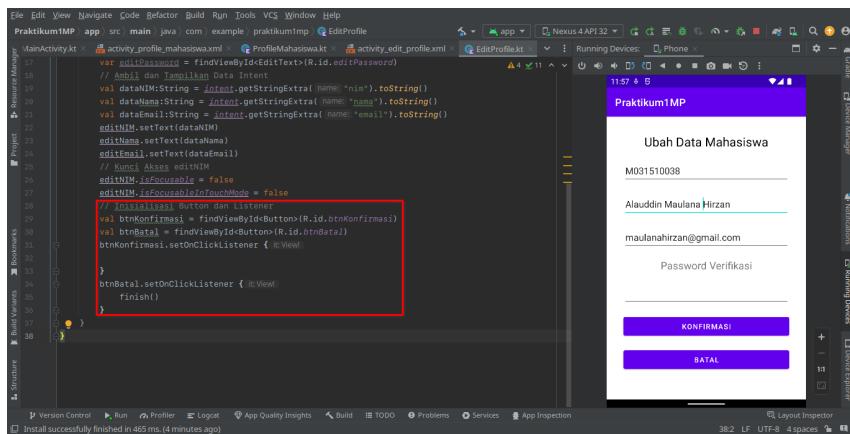


7. Berikutnya tambahkan kode inisialisasi Button dan fungsi Listener di masing-masing tombol. Perhatikan kode berikut

Potongan Kode

```

// Inisialisasi Button dan Listener
val btnKonfirmasi = findViewById<Button>(R.id.btnKonfirmasi)
val btnBatal = findViewById<Button>(R.id.btnBatal)
btnKonfirmasi.setOnClickListener {
    ...
}
btnBatal.setOnClickListener {
    finish()
}
```

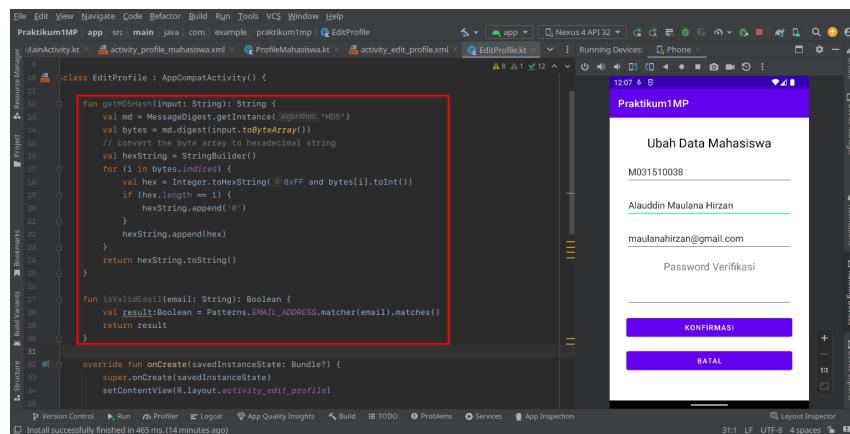


8. Berikutnya masukkan kode untuk mendukung perubahan data seperti **Hashing MD5** dan **E-Mail Checker**. Letakkan dibawah baris kode **Class**. Perhatikan contoh berikut:

Potongan Kode

```
fun getMD5Hash(input: String): String {
    val md = MessageDigest.getInstance("MD5")
    val bytes = md.digest(input.toByteArray())
    // convert the byte array to hexadecimal string
    val hexString = StringBuilder()
    for (i in bytes.indices) {
        val hex = Integer.toHexString(0xFF and bytes[i].toInt())
        if (hex.length == 1) {
            hexString.append('0')
        }
        hexString.append(hex)
    }
    return hexString.toString()
}

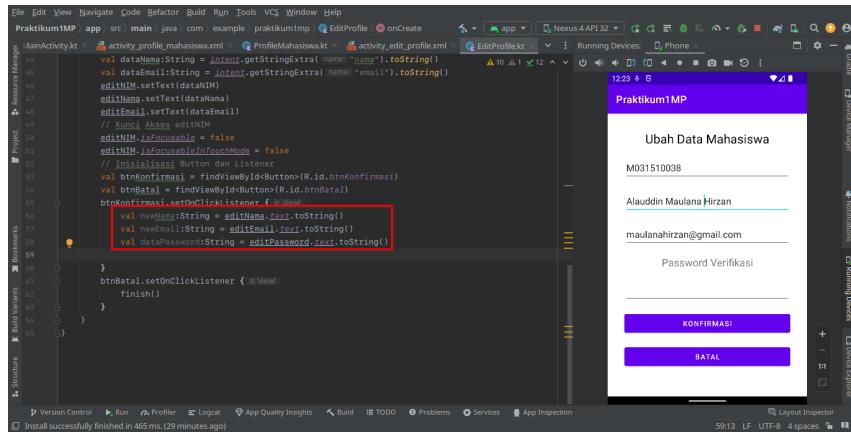
fun isValidEmail(email: String): Boolean {
    val result:Boolean = Patterns.EMAIL_ADDRESS.matcher(email).matches()
    return result
}
```



9. Pindah ke bagian **btnKonfirmasi** dan tambahkan kode berikut untuk mengambil data baru. Perhatikan contoh kode dan gambar:

Potongan Kode

```
val newName:String = editNama.text.toString()
val newEmail:String = editEmail.text.toString()
val dataPassword:String = editPassword.text.toString()
```



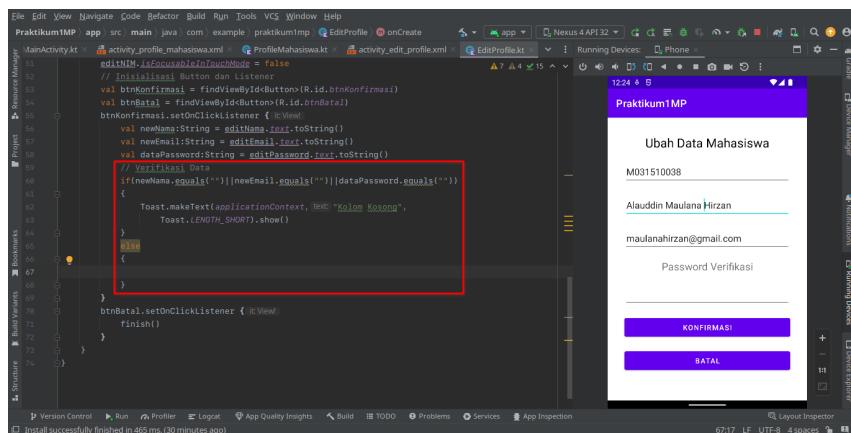
10. Lalu tambahkan kode untuk verifikasi data baru. Perhatikan contoh:

Potongan Kode

```

// Verifikasi Data
if(newName.equals("")||newEmail.equals("")||dataPassword.equals(""))
{
    Toast.makeText(getApplicationContext,"Kolom Kosong",
        Toast.LENGTH_SHORT).show()
}
else
{
}

```



11. Kemudian tambahkan kode verifikasi E-Mail tepat di dalam ELSE

Potongan Kode

```

if(isValidEmail(newEmail))
{
}
else
{
    Toast.makeText(getApplicationContext,"E-Mail Tidak Valid",
        Toast.LENGTH_SHORT).show()
}

```

```

    // Verifikasi Data
    if(newName.equals("")||newEmail.equals("")||editPassword.equals(""))
    {
        Toast.makeText(getApplicationContext(), text: "Kolom Kosong",
        Toast.LENGTH_SHORT).show()
    }
    else
    {
        if(isValidEmail(newEmail))
        {
        }
        else
        {
            Toast.makeText(getApplicationContext(), text: "E-Mail Tidak Valid",
            Toast.LENGTH_SHORT).show()
        }
    }
    btnBatal.setOnClickListener { it:View?
    finish()
}

```

12. Kode hampir siap. Namun sebelum itu, tambahkan kode **late init** lagi tepat di bawah baris kode **Class**

Potongan Kode
private lateinit var databaseHelper: DBHelper

```

    package com.example.praktikum1mp

import ...

class EditProfile : AppCompatActivity() {
    private lateinit var databaseHelper: DBHelper

    fun getMD5Hash(input: String): String {
        val md = MessageDigest.getInstance(algorithm: "MD5")
        val bytes = md.digest(input.toByteArray())
        // convert the byte array to hexadecimal string
        val hexString = StringBuilder()
        for (i in bytes.indices) {
            val hex = Integer.toHexString(0xFF and bytes[i].toInt())
            if (hex.length == 1) {
                hexString.append("0")
            }
            hexString.append(hex)
        }
        return hexString.toString()
    }

    fun isValidEmail(email: String): Boolean {
        val result:Boolean = Patterns.EMAIL_ADDRESS.matcher(email).matches()
        return result
    }
}

```

13. Kemudian tambahkan kode Inisialisasi DB tepat di bawah kode inisialisasi **Button**. Perhatikan gambar

Potongan Kode
databaseHelper = DBHelper(this)

```

    val dataNameString = intent.getStringExtra(name: "name").toString()
    val dataEmailString = intent.getStringExtra(name: "email").toString()
    editName.setText(dataNameString)
    editEmail.setText(dataEmailString)
    editEmail.setCursorVisible(true)
    editName.requestFocus()
    editName.isFocusable = false
    editName.isFocusableInTouchMode = false
    // Inisialisasi Button dan Listener
    val btnKonfirmasi = findViewById(R.id.btnKonfirmasi)
    val btnBatal = findViewById(R.id.btnBatal)

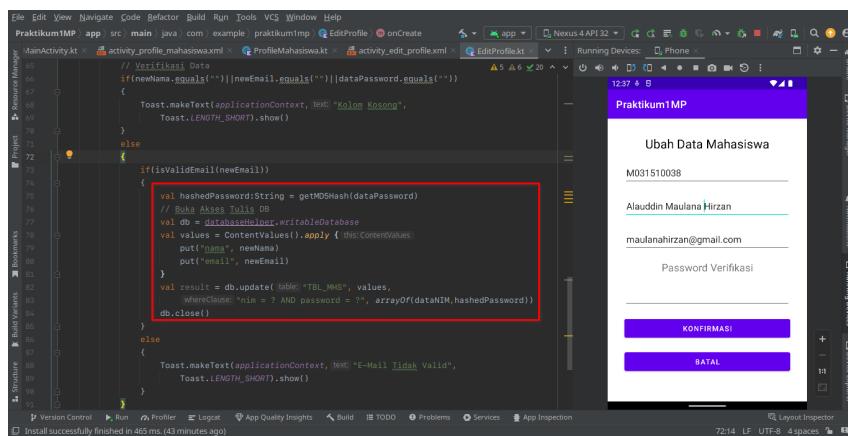
    databaseHelper = DBHelper(context: this)

```

14. Kini sudah siap berikutnya menambahkan kode inti **IF** di tombol **btnKonfirmasi** sebelumnya

Potongan Kode

```
val hashedPassword:String = getMD5Hash(dataPassword)
// Buka Akses Tulis DB
val db = databaseHelper.writableDatabase
val values = ContentValues().apply {
    put("nama", newNama)
    put("email", newEmail)
}
val result = db.update("TBL_MHS", values,
    "nim = ? AND password = ?", arrayOf(dataNIM,hashedPassword))
db.close()
```



15. Terakhir tambahkan kode pengecekan update tepat di bawah kode sebelumnya. Perhatikan contoh:

Potongan Kode

```
// Cek Sukses Kueri
if(result > 0)
{
    Toast.makeText(getApplicationContext(),"Update Berhasil",
    Toast.LENGTH_SHORT).show()
    finish()
}
else
    Toast.makeText(getApplicationContext(),"Update Gagal",
    Toast.LENGTH_SHORT).show()
```

The screenshot shows the Android Studio interface with the code editor open to MainActivity.kt. The code handles updating a user's profile in a database. It checks if the NIM and password are valid, updates the database, and then displays a success or failure toast message. The application is running on an emulator named 'Nexus 4 API 32'.

```
    put("email", newEmail)
    val result = db.update(table: "TBL_HHS", values,
        whereClause: "nim = ? AND password = ?", arrayOf(dataNIM, hashedPassword))
    db.close()
    // Cek Sukses Kueri
    if(result > 0)
    {
        Toast.makeText(applicationContext, text: "Update Berhasil",
            Toast.LENGTH_SHORT).show()
        finish()
    }
    else
        Toast.makeText(applicationContext, text: "Update Gagal",
            Toast.LENGTH_SHORT).show()
    }
    btnBatal.setOnClickListener { it:View
        finish()
    }
}
```

16. Jalankan Emulator dan Aplikasi. Dan Ubah Nama / E-Mail. Pastikan untuk **LOGOUT** setelah berhasil mengubah Data.

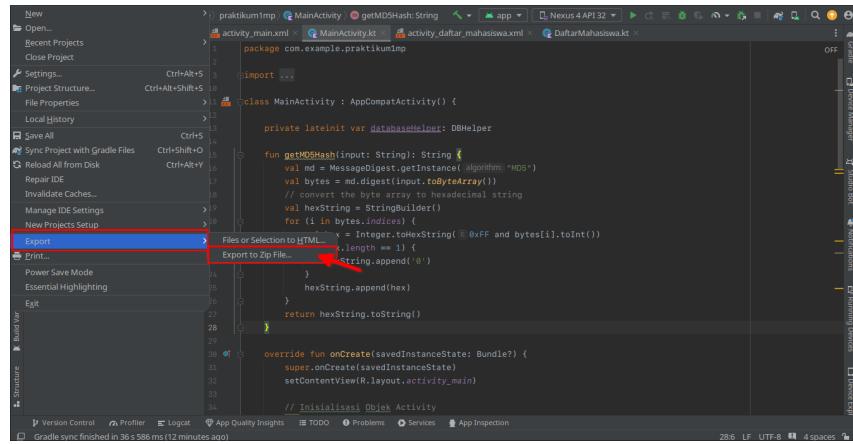
The screenshot shows the Android Studio interface with the code editor open to MainActivity.kt. The code is identical to the previous one, but the toast message 'Update Berhasil' is highlighted with a red rectangle. The application is running on an emulator named 'Nexus 4 API 32'.

```
    put("email", newEmail)
    val result = db.update(table: "TBL_HHS", values,
        whereClause: "nim = ? AND password = ?", arrayOf(dataNIM, hashedPassword))
    db.close()
    // Cek Sukses Kueri
    if(result > 0)
    {
        Toast.makeText(applicationContext, text: "Update Berhasil",
            Toast.LENGTH_SHORT).show()
        finish()
    }
    else
        Toast.makeText(applicationContext, text: "Update Gagal",
            Toast.LENGTH_SHORT).show()
    }
    btnBatal.setOnClickListener { it:View
        finish()
    }
}
```

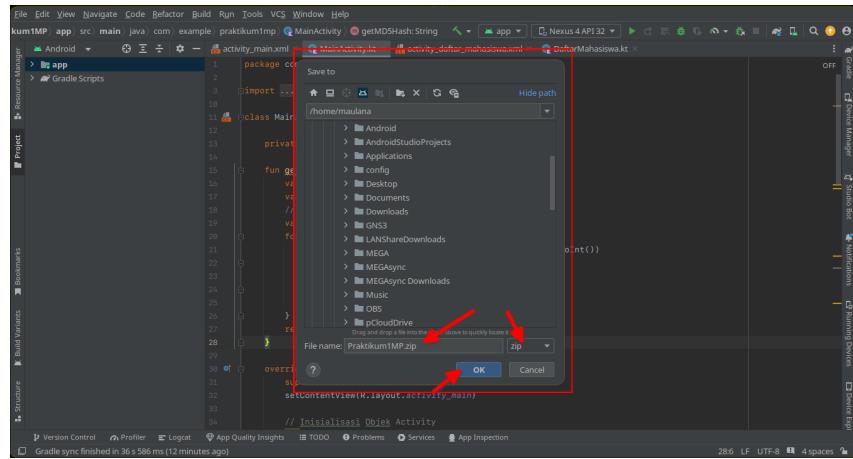
The screenshot shows the Android Studio interface with the code editor open to MainActivity.kt. The code is identical to the previous ones. The updated email address 'maulanahirzan@usm.ac.id' is highlighted with a red rectangle in the application's UI. The application is running on an emulator named 'Nexus 4 API 32'.

```
    put("email", newEmail)
    val result = db.update(table: "TBL_HHS", values,
        whereClause: "nim = ? AND password = ?", arrayOf(dataNIM, hashedPassword))
    db.close()
    // Cek Sukses Kueri
    if(result > 0)
    {
        Toast.makeText(applicationContext, text: "Update Berhasil",
            Toast.LENGTH_SHORT).show()
        finish()
    }
    else
        Toast.makeText(applicationContext, text: "Update Gagal",
            Toast.LENGTH_SHORT).show()
    }
    btnBatal.setOnClickListener { it:View
        finish()
    }
}
```

17. Untuk mengirimkan hasilnya, klik **File**, pilih **Export**, dan pilih **Export to ZIP**.



18. Pilih lokasi yang mudah diingat, pastikan nama dan format ZIP sudah sesuai, dan klik **OK**



Bab 8

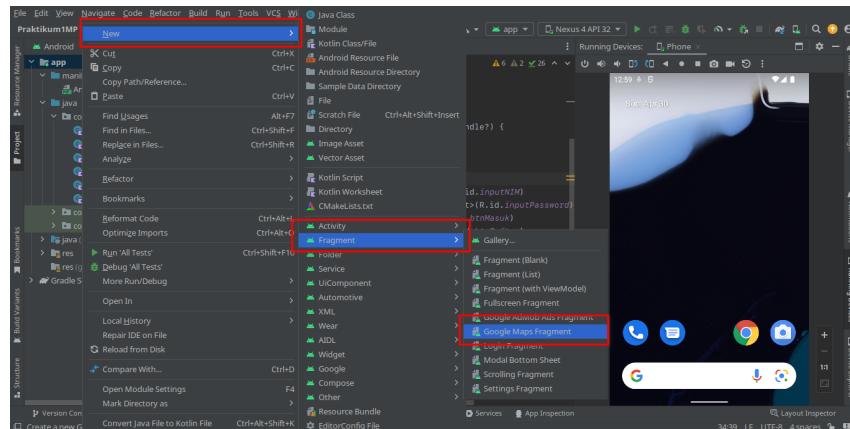
Praktikum 8

8.1 Google Map

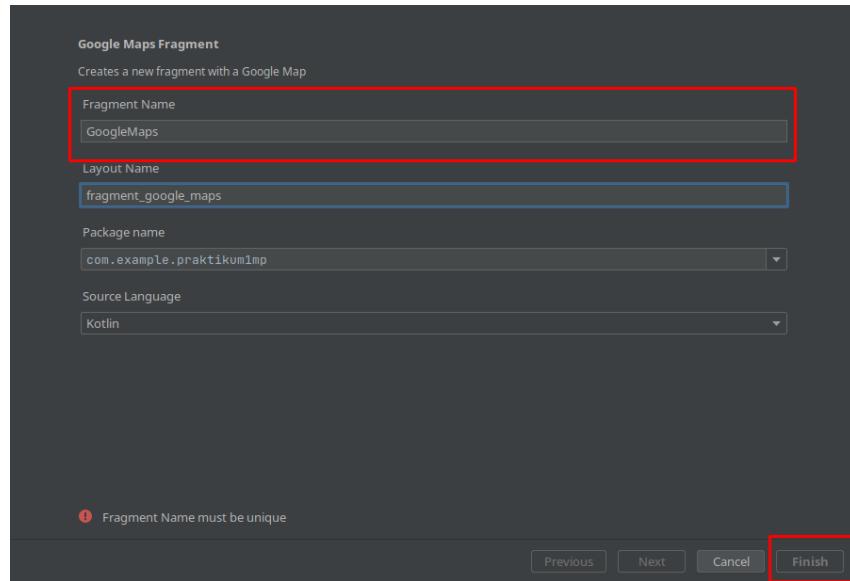
Di bagian ini mahasiswa diajarkan bagaimana cara menambahkan fungsi **Google Maps** di bagian **Profile Mahasiswa**. Mahasiswa diwajibkan untuk menyelesaikan Praktikum 7

8.2 Tutorial

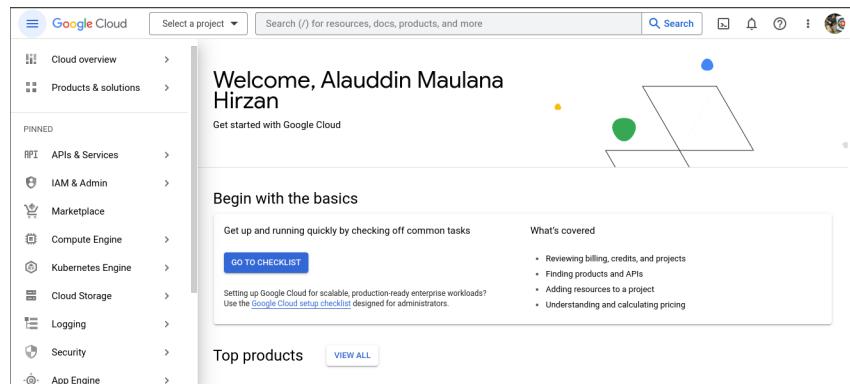
1. Buka Android Studio dan buatlah **Fragment** bukan **Activity** dengan melakukan klik Kanan **app Project** → pilih **New** → pilih **Fragment** dan Klik **Google Maps Fragment**



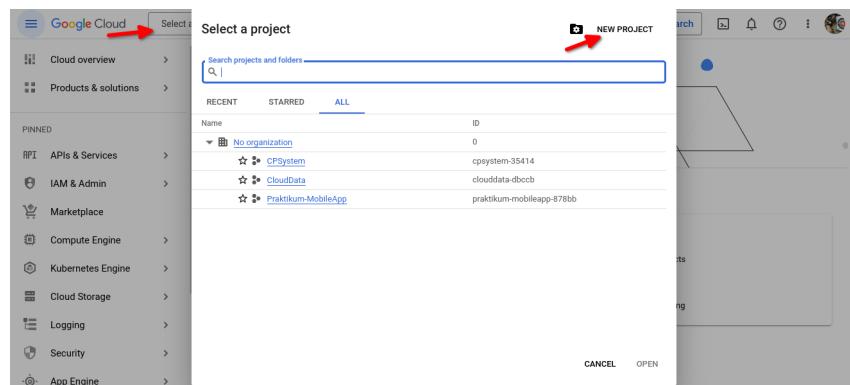
2. Beri nama **GoogleMaps**, dan klik **Finish**



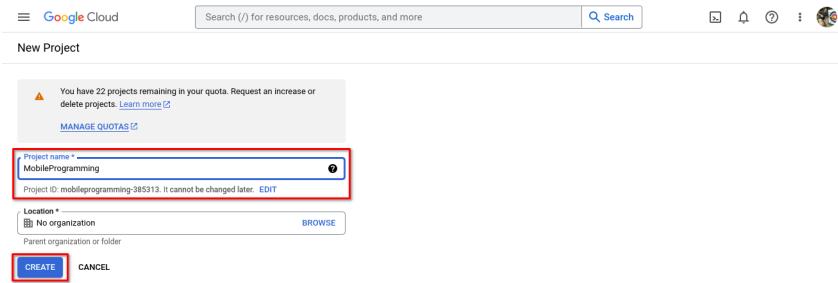
3. Tunggu beberapa saat hingga **Android Studio** selesai melakukan loading.
4. Setelah itu buka website <https://console.cloud.google.com>. Login dengan akun Google pribadi. Jika sudah pastikan berada di halaman Dasbornya seperti gambar berikut:



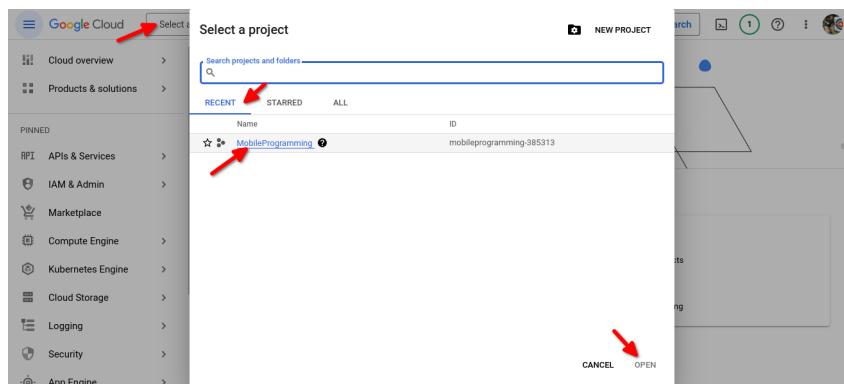
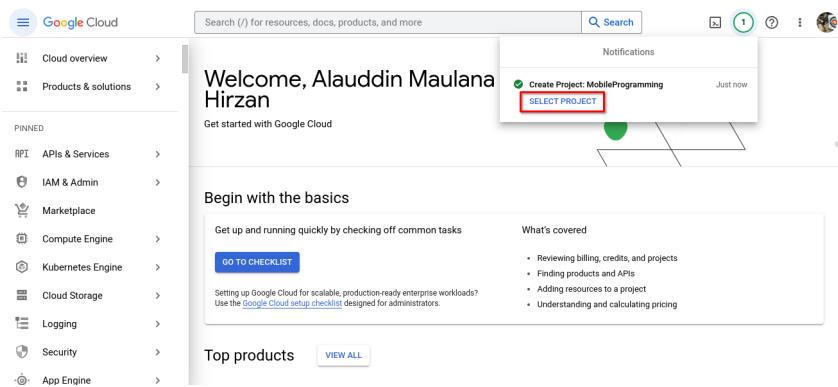
5. Buatlah projek baru dengan klik **Select Project** di bagian atas, dan pilih **New Project**



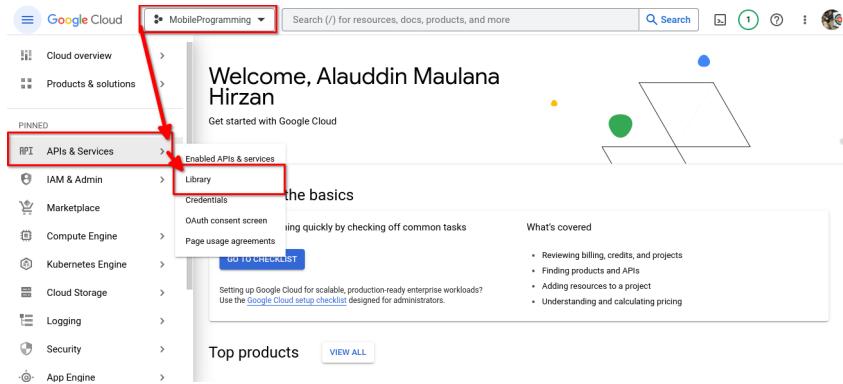
6. Beri nama projek **MobileProgramming** tanpa spasi, dan klik **Create**



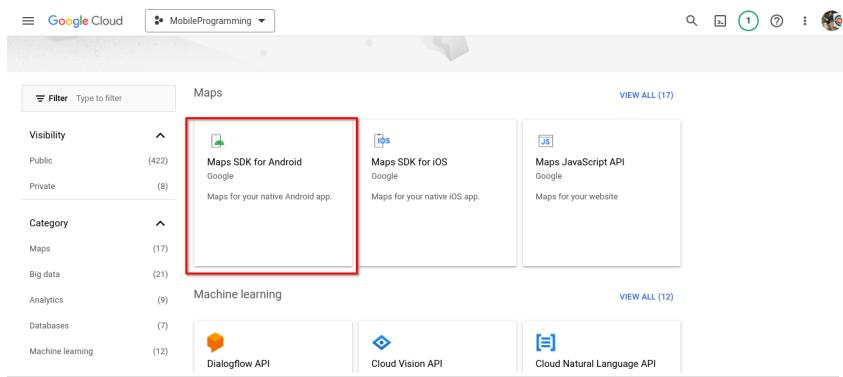
7. Tunggu Google membuat projeknya, dan pastikan projek yang sudah dibuat sudah terpilih. Klik **Select Project** di bagian notifikasi atau Dasbor



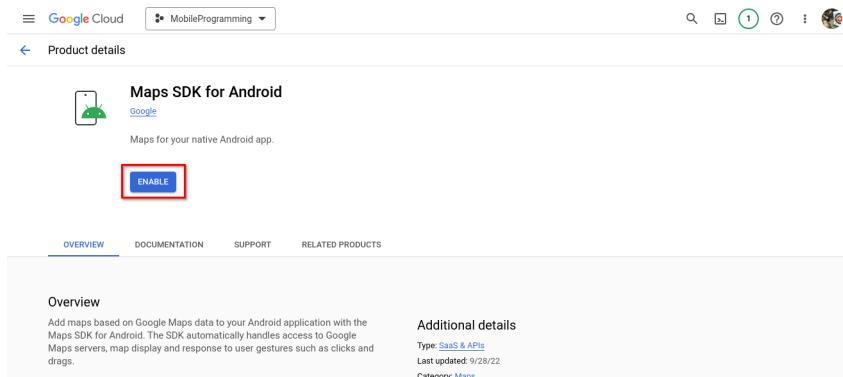
8. Berikutnya adalah membuka akses projek ke Google Apps Maps. Sesudah memilih projek, di bagian panel kiri pilih **APIs & Services**, lalu pilih **Library**



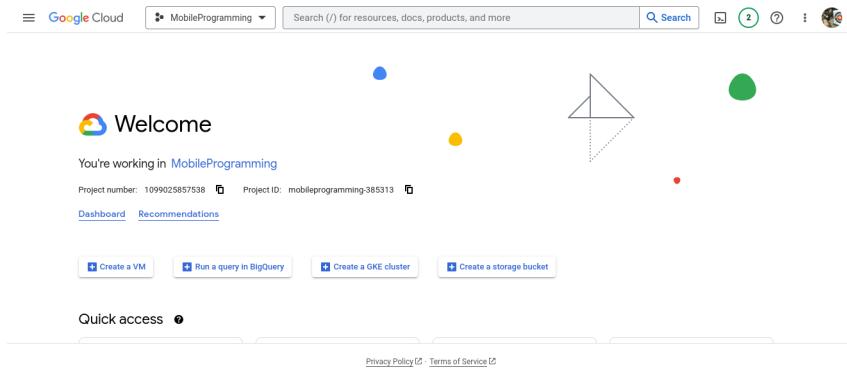
9. Halaman akan dialihkan ke daftar Apps Google, pilih Maps SDK for Android



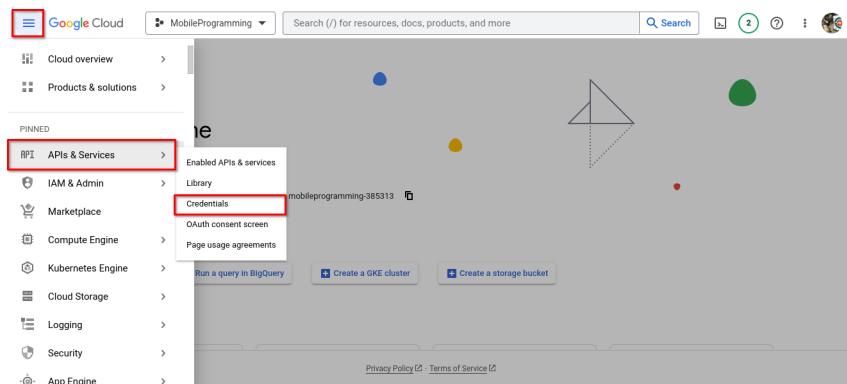
10. Klik Enable untuk mengizinkan projek mengakses Maps SDK for Android



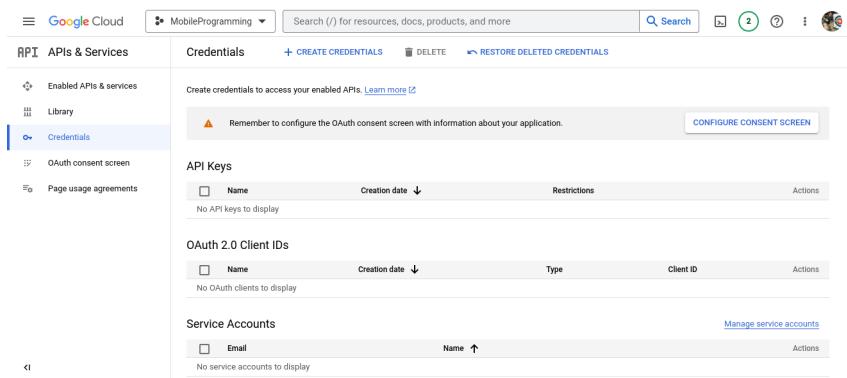
11. Tunggu beberapa saat hingga halaman berubah. Jika tidak refresh halaman.
12. Jika Google meminta pendaftaran pembayaran / **Billing** klik Cancel. Atau klik **Logo Google Cloud**
13. Kembali ke halaman Dasbor projek



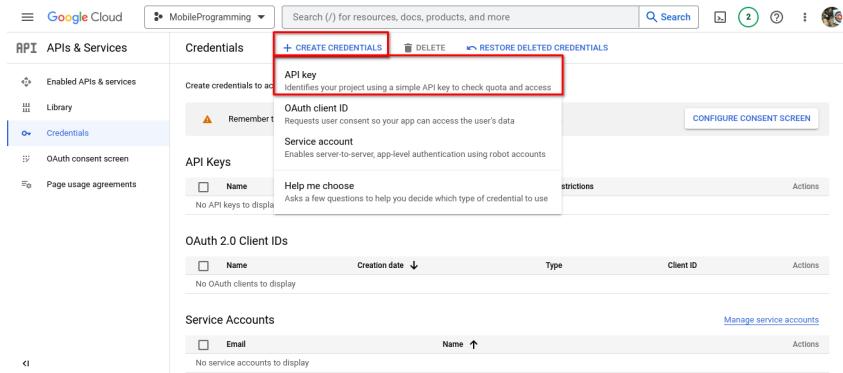
14. Klik 3 Garis Hitam di pojok kiri atas, pilih APIs & Services, pilih Credentials



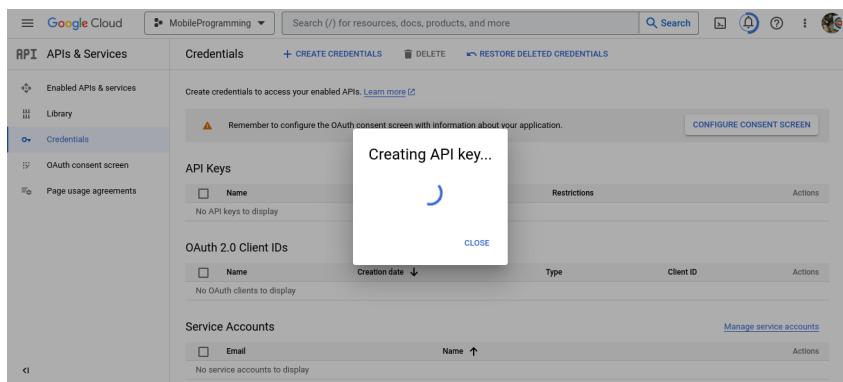
15. Halaman akan berpindah, dan di sinilah mahasiswa akan membuat API Keys untuk Android Studio



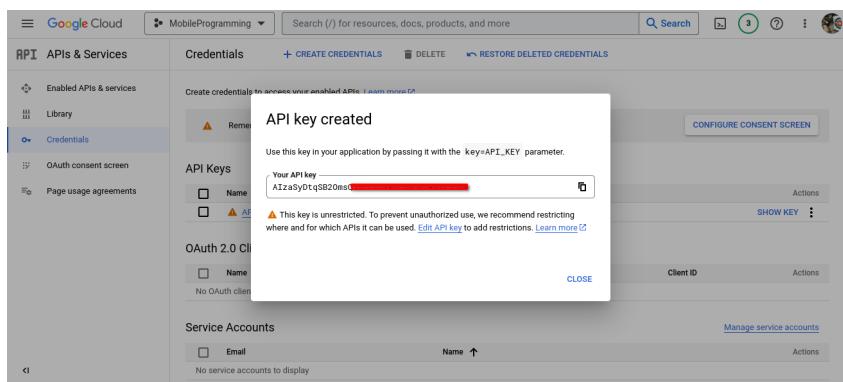
16. Untuk membuat API Keys, klik Create Credentials, pilih API Key



17. Google akan membuat kunci, tunggu hingga selesai



18. Kunci akan ditampilkan, dan klik **Copy**



19. Buka Android Studio kembali, dan buka file **Android Manifest** atau klik **Project** → folder **manifests** → **AndroidManifest.xml**. Cari garis kode seperti di gambar berikut:

```

<manifest ...>
    ...
    <application ...>
        ...
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="YOUR_API_KEY" />
        ...
    </application>
</manifest>

```

20. Tempel API Key yang sudah dibuat tadi menggantikan YOUR_API_KEY dalam tanda petik:

```

<manifest ...>
    ...
    <application ...>
        ...
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="1IzaSy0rq8R20msQ5z..." />
        ...
    </application>
</manifest>

```

21. Berikutnya adalah menghapus semua isi dari GoogleMaps.kt dan hanya meninggalkan baris kode Class saja. Lihat gambar:

```

package com.example.praktikum1p
import androidx.fragment.app.Fragment
class GoogleMaps : Fragment() {
}

```

22. Ubah sintaks Fragment menjadi DialogFragment(), OnMapReadyCallback. Perhatikan contoh gambar:

The screenshot shows the Android Studio interface with the code editor open. The class `GoogleMaps` is defined as follows:

```
package com.example.praktikum1mp

import androidx.fragment.app.Fragment

class GoogleMaps : DialogFragment(), OnMapReadyCallback {
```

The line `import androidx.fragment.app.Fragment` is highlighted with a red box, indicating an error.

23. Perbaiki Error dengan Import ALT+ENTER

The screenshot shows the same code editor after performing an import. The imports are now correctly added:

```
package com.example.praktikum1mp

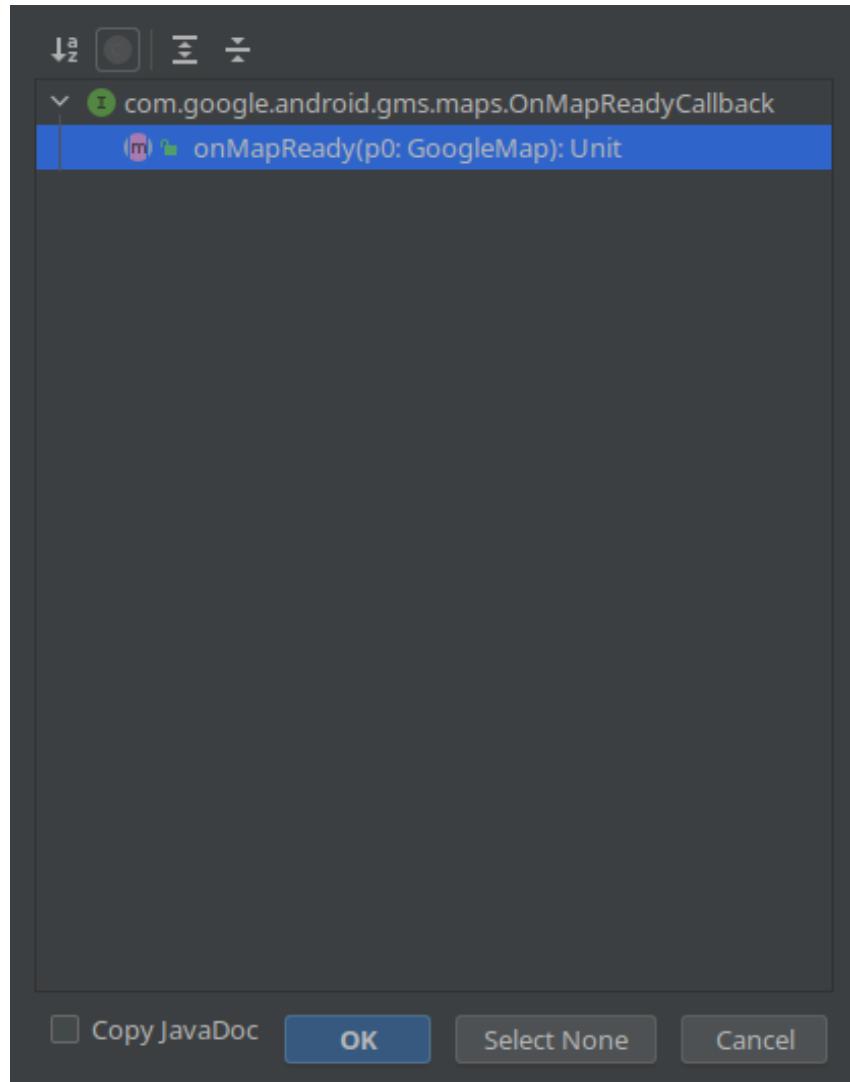
import androidx.fragment.app.DialogFragment
import com.google.android.gms.maps.OnMapReadyCallback
```

The code editor shows the imports highlighted with a red box.

24. Sekarang baris kode **class GoogleMaps** menjadi **Error**. Klik kanan pilih **Show Context Actions**, pilih **Implement Members**, dan klik **OK**

The screenshot shows the context menu for the `GoogleMaps` class. The option `Implement members` is selected and highlighted with a red box.

- Implement members
- Make 'GoogleMaps' abstract
- Add Parcelable Implementation
- Create test

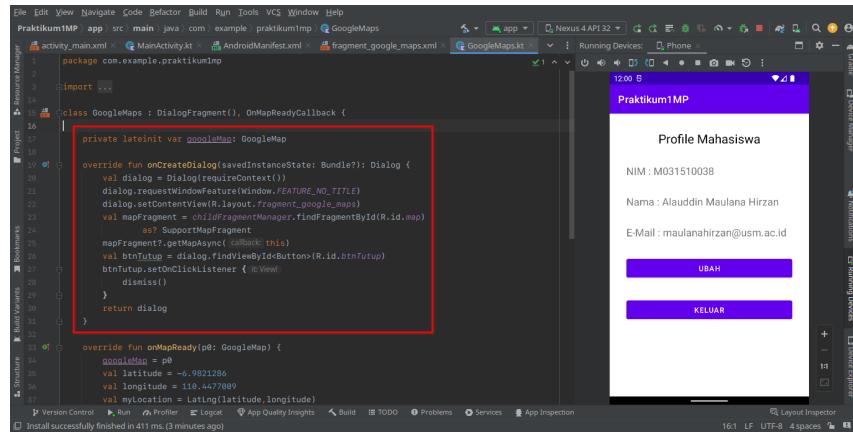


25. Berikutnya memrogram file tersebut, letakkan kode berikut ini tepat di bawah baris kode **class GoogleMaps**

Potongan Kode

```
private lateinit var googleMap: GoogleMap

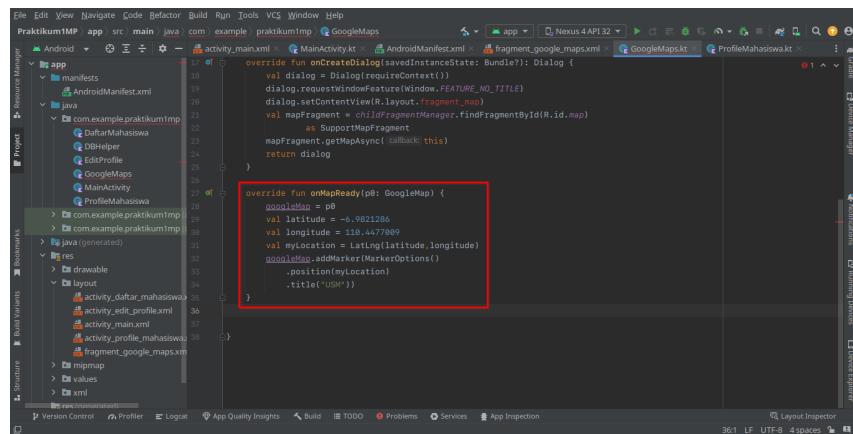
override fun onCreateDialog(savedInstanceState: Bundle?): Dialog {
    val dialog = Dialog(requireContext())
    dialog.requestWindowFeature(Window.FEATURE_NO_TITLE)
    dialog.setContentView(R.layout.fragment_map)
    val mapFragment = childFragmentManager.findFragmentById(R.id.map)
        as? SupportMapFragment
    mapFragment?.getMapAsync(this)
    val btnTutup = dialog.findViewById<Button>(R.id.btnTutup)
    btnTutup.setOnClickListener {
        dismiss()
    }
    return dialog
}
```



26. ABAIKAN ERROR R.layout.fragment_map. Lanjutkan kode dengan mengubah isi onMapReady dengan kode berikut

Potongan Kode

```
googleMap = p0
val latitude = -6.9821286
val longitude = 110.4477009
val myLocation = LatLng(latitude,longitude)
googleMap.addMarker(MarkerOptions()
    .position(myLocation)
    .title("USM"))
googleMap.uiSettings.isZoomControlsEnabled = true
googleMap.uiSettings.isMyLocationButtonEnabled = true
googleMap.addMarker(MarkerOptions().position(myLocation))
googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(myLocation, 15f))
```



27. Bagian berikutnya, lanjutkan kode dengan kode berikut

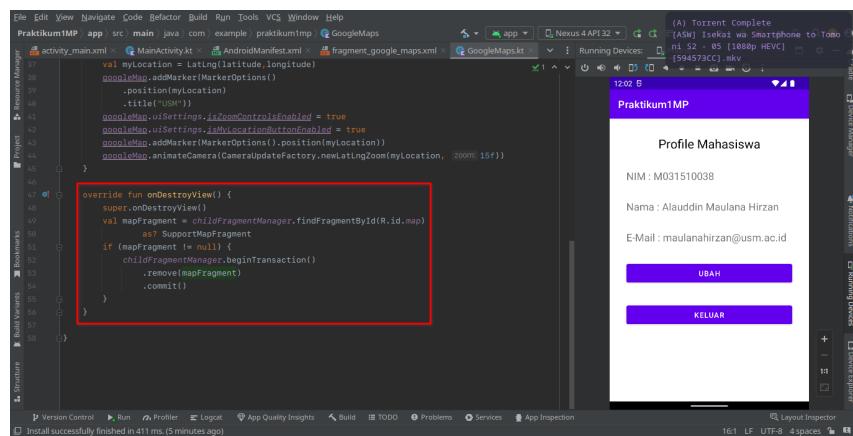
Potongan Kode

```

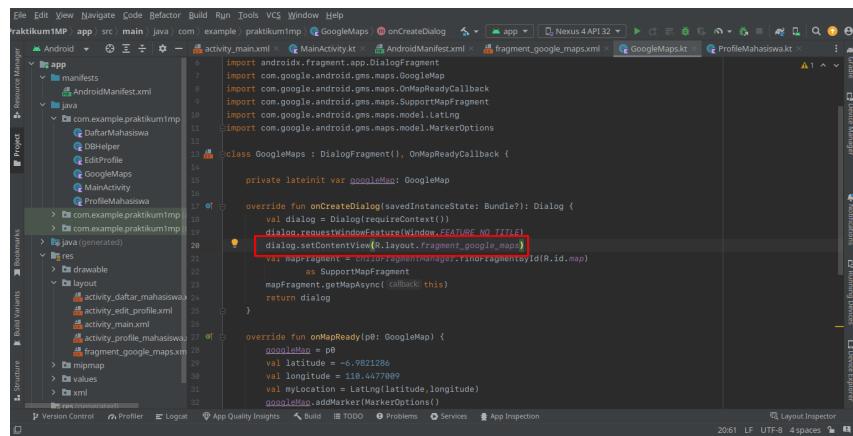
override fun onDestroyView() {
    super.onDestroyView()

    val mapFragment = childFragmentManager.findFragmentById(R.id.map)
        as? SupportMapFragment
    if (mapFragment != null) {
        childFragmentManager.beginTransaction()
            .remove(mapFragment)
            .commit()
    }
}

```



28. Untuk memperbaiki **Error fragment_map**. Ubah kode tersebut menjadi **fragment_google_maps**



29. Berikutnya ubah isi file **fragment_google_maps.xml** dengan membungkus kode yang ada ke dalam **LinearLayout**. Perhatikan kode dan Gambar:

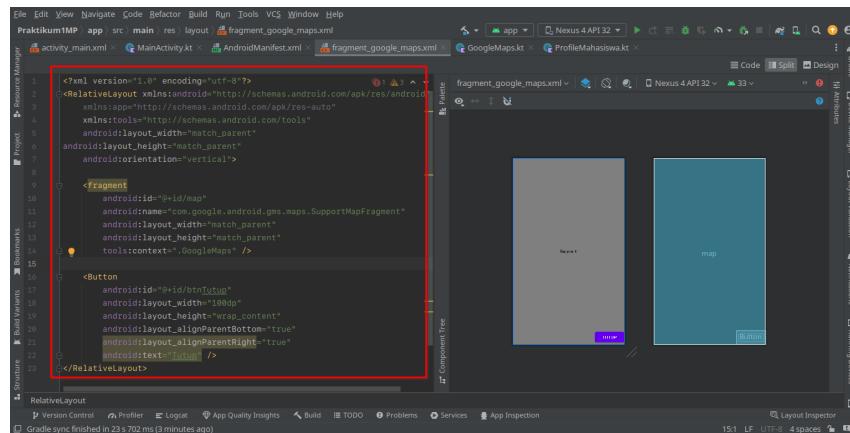
Potongan Kode

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

    -- Kode Fragment Asli --

    <Button
        android:id="@+id	btnTutup"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:text="Tutup" />

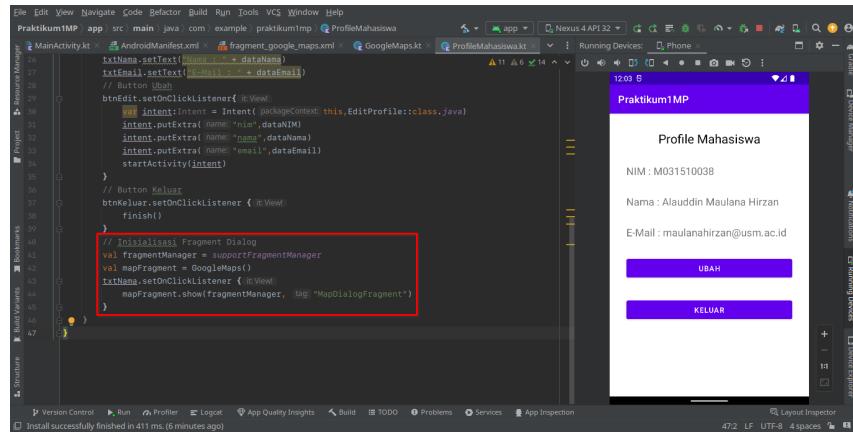
</RelativeLayout>
```



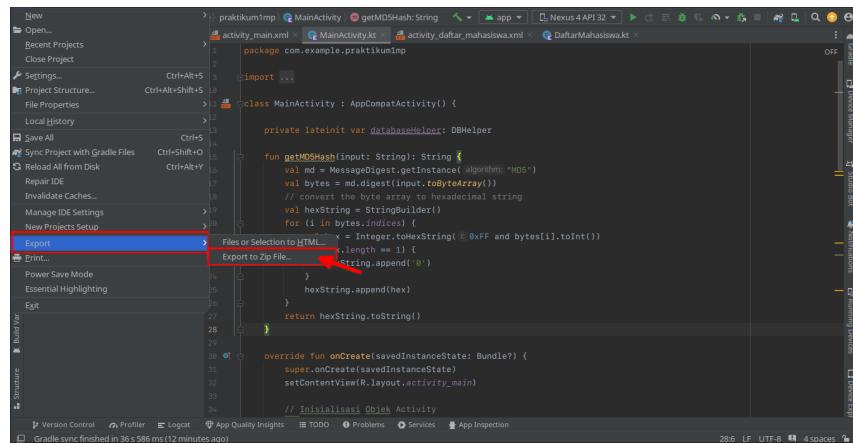
30. Jika sudah, buka file **ProfileMahasiswa.kt** dan tambahkan kode berikut tepat di bawah kode **btnKeluar**. Perhatikan kode dan gambar:

Potongan Kode

```
// Inisialisasi Fragment Dialog
val fragmentManager = supportFragmentManager
val mapFragment = GoogleMaps()
txtNama.setOnClickListener {
    mapFragment.show(fragmentManager, "MapDialogFragment")
}
```



31. Jalankan Emulator dan Aplikasi. Saat ini aplikasi hanya bisa membuka Map tanpa Zoom Auto dan Marker, dan Aplikasi akan Error ketika membuka map kedua kalinya
32. Untuk mengirimkan hasilnya, klik **File**, pilih **Export**, dan pilih **Export to ZIP**.



33. Pilih lokasi yang mudah diingat, pastikan nama dan format ZIP sudah sesuai, dan klik **OK**

