

MODUL PRAKTIKUM

PEMROGRAMAN

PERANGKAT BERGERAK



Disusun oleh:
ARIF RIFAI DWIYANTO

PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BHAYANGKARA JAKARTA RAYA
2023

DAFTAR ISI

PENDAHULUAN

PRAKTIKUM 1

PRAKTIKUM 2

PRAKTIKUM 3

TUGAS KELOMPOK

PENGANTAR

PENDAHULUAN

Praktikum yang dilakukan adalah membuat aplikasi “Panic Button” mobile.

Deskripsi permasalahan yang akan dikerjakan:

Keadaan darurat bisa terjadi kapan saja di mana saja. Jika terjadi keadaan darurat maka kondisi tersebut perlu diinformasikan kepada pihak yang berwenang untuk dilakukan penanganan/mitigasi. Salah satu sarana untuk menginformasikan keadaan darurat adalah dengan menggunakan tombol alarm darurat atau *panic button*.

Panic button adalah salah satu teknologi keamanan yang mempunyai banyak nama lain seperti alarm darurat, emergency signal, SOS alarm, personal alarm, alert button, maupun sirene. Salah satu jenis safety device ini banyak digunakan di berbagai industri maupun perkantoran. Panic button memiliki fungsi untuk membunyikan alarm dalam keadaan darurat (*emergency*) di saat seseorang membutuhkan bantuan dengan segera.



Gambar 1. Alarm Darurat di Gedung SS Ubhara dipasang di samping APAR dan mesin absensi

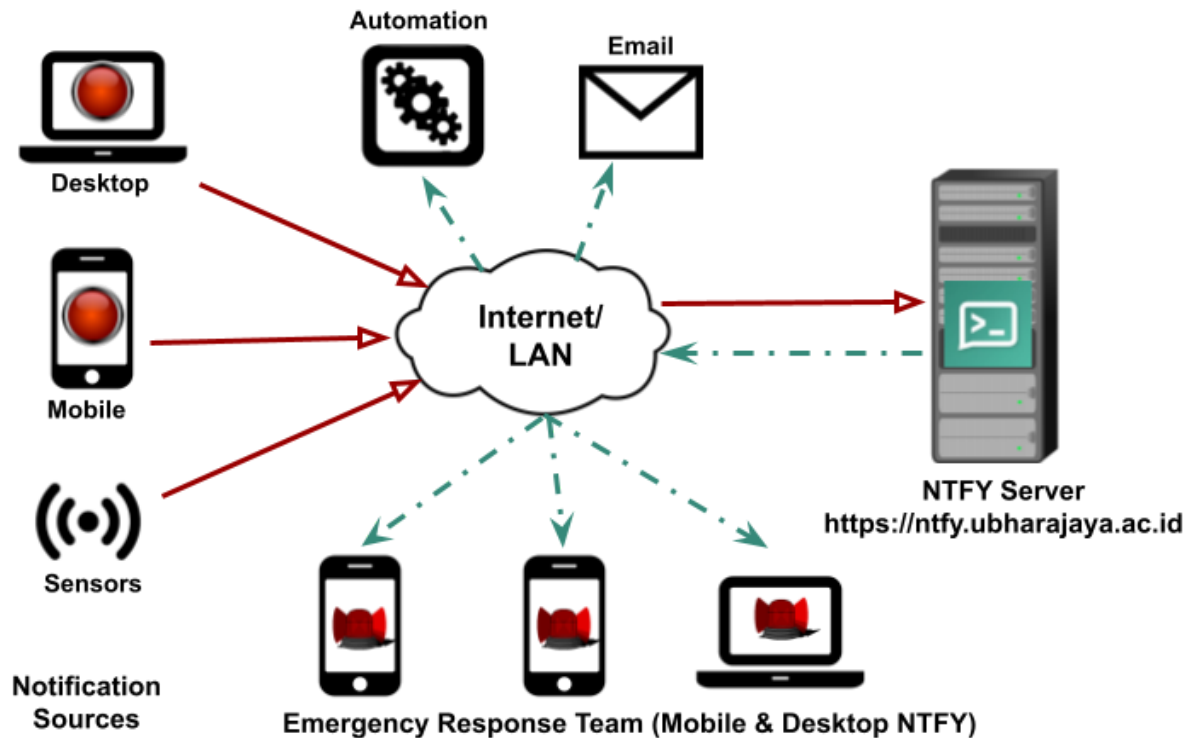
Selain alarm sirine yang dapat berbunyi keras, tombol panik juga dapat mengirimkan alarm senyap (silent alarm) yang memberi peringatan kepada petugas keamanan, manajemen, atau layanan darurat jika sedang terjadi suatu insiden. Silent alarm penting karena pelapor dapat meminta bantuan tanpa diketahui oleh penjahat.

Teknologi keamanan saat ini bahkan sudah mengembangkan personal panic button berupa hardware key fob yang terhubung dengan smartphone.



Gambar 2. Smart SOS Button

Dalam tugas praktikum ini akan coba dibuat aplikasi panic button berbasis Android. Ketika panic button ditekan maka aplikasi akan mengirimkan silent alarm melalui sarana push notification NTFY <https://ntfy.sh/>



Gambar 3. Proses Pengiriman Alert dan Notifikasi

<https://play.google.com/store/apps/details?id=io.heckel.ntfy>

Server NTFY yang akan digunakan beralamat di <https://ntfy.ubharajaya.ac.id/> dengan topik “panic-button” <https://ntfy.ubharajaya.ac.id/panic-button>

Pengembangan lebih lanjut:

- layar akan memunculkan animasi flashing sirine berwarna merah/biru
- smartphone dapat mengeluarkan bunyi sirine jika diinginkan
- lokasi pelapor akan dikirimkan melalui ntfy
- pelapor bisa mengambil gambar, suara, atau video kejadian dan mengirimkan melalui push notification NTFY
- dapat subscribe push notification NTFY untuk pemberitahuan kondisi bahaya/darurat.
- Aplikasi memberikan informasi/petunjuk tindakan jika terjadi keadaan bahaya/darurat sesuai lokasi gedung

- Security checklist
- Security checkpoint
- Integrasi dengan MacroDroid atau Tasker
- Koneksi dengan perangkat bluetooth
- dll

A. Sesi Praktikum Pertama

Prasyarat

- Sudah menginstal android studio
- Sudah pernah membuat aplikasi hello world dengan android studio
- Sudah pernah membuat “aplikasi kartu ucapan selamat”

Persiapan

1. Install Android studio di laptop <https://developer.android.com/studio> dan coba buat aplikasi hello world bagi yang baru pertama kali menjalankan android studio.
2. Install aplikasi scrcpy. Aplikasi scrcpy digunakan untuk mirroring tampilan dari handset android ke komputer.
 - a. Download <https://github.com/Genymobile/scrcpy/releases>.
 - b. Aktifkan USB Debugging di HP Android
 - c. hubungkan HP Android dengan komputer menggunakan kabel data
 - d. jalankan scrcpy maka tampilan HP android akan muncul di komputer
3. Install aplikasi NTFY dari playstore atau dari F-Droid
 - a. Download dari
 - i. <https://f-droid.org/packages/io.heckel.ntfy/> atau
 - ii. <https://play.google.com/store/apps/details?id=io.heckel.ntfy>
 - b. subscribe topic <https://ntfy.ubharajaya.ac.id/panic-button> jika ada notifikasi ke topik ini maka HP akan menerima notifikasi.
 - c. Selain menginstall aplikasi NTFY bisa juga membuka alamat <https://ntfy.ubharajaya.ac.id/panic-button> dengan browser.

Pembuatan Aplikasi

1. Buat Satu Project dengan judul “Panic-button”
2. Buat Satu Activity dan layout yang berisi button
3. Jika ada event klik button maka akan dimunculkan toast alert dan mengirimkan post/get http request ke <https://ntfy.ubharajaya.ac.id/panic-button>
4. Cek notifikasi apakah masuk baik melalui web atau aplikasi android

5. Sebagai starter dapat digunakan aplikasi dice-roller atau dessert.

Berikut adalah langkah-langkah umum menggunakan bahasa pemrograman Kotlin dan Android Studio:

Buka Android Studio:

Pastikan Anda sudah menginstal Android Studio di komputer Anda.

Buat Proyek Baru:

Buka Android Studio dan buat proyek baru.

Pilih **"Empty Views Activity"** agar proyek Anda memiliki satu view activity.

Desain Layout:

Buka file **activity_main.xml** di folder res/layout.

Tambahkan tombol ke layout XML, seperti contoh berikut:

xml

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Kirim Alert"
        android:layout_centerInParent="true"/>
</RelativeLayout>
```

Tambahkan Kode Kotlin:

Buka file **MainActivity.kt** di folder **src**.

Tambahkan kode berikut untuk menangani klik tombol dan mengirim HTTP POST request:

kotlin

```
import android.os.AsyncTask
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import java.io.OutputStream
import java.net.HttpURLConnection
import java.net.URL

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val button = findViewById<Button>(R.id.button)
        button.setOnClickListener {
            // Panggil metode untuk mengirim HTTP POST request
            SendPostRequest().execute()
        }
    }

    private class SendPostRequest : AsyncTask<Void, Void, Void>() {
        override fun doInBackground(vararg params: Void?): Void? {
            try {
                val url =
URL("https://ntfy.ubharajaya.ac.id/panic-button")
                val connection = url.openConnection() as
HttpURLConnection
                connection.requestMethod = "POST"

                // Tambahkan parameter POST jika diperlukan
                // Misalnya, jika Anda ingin mengirim data JSON,
gunakan:
                // connection.outputStream.use { os ->
os.write("json_data".toByteArray()) }
```

```

        val responseCode = connection.responseCode
        // Proses responseCode atau response body jika
diperlukan

        connection.disconnect()
    } catch (e: Exception) {
        e.printStackTrace()
    }
    return null
}
}
}

```

Pastikan untuk menyesuaikan kode sesuai kebutuhan, seperti menambahkan parameter POST atau menangani responseCode dan response body.

Buka Izin untuk akses Internet:

Pastikan Anda telah menambahkan izin internet di file AndroidManifest.xml:

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

Uji Coba:

Jalankan aplikasi di emulator virtual device atau perangkat fisik Anda.

Tekan tombol, dan aplikasi seharusnya mengirim HTTP POST request ke alamat yang ditentukan. dan muncul notifikasi “triggered” di topic panic-button NTFY.

Tambahkan informasi header dan content untuk NTFY

```

// Atur header request dengan informasi yang diinginkan
connection.setRequestProperty("Content-Type", "text/plain")
connection.setRequestProperty("Title", "Alert")
connection.setRequestProperty("Priority", "urgent")
connection.setRequestProperty("Tags", "alert warning,rotating_light")

```

```
// Tambahkan konten ke permintaan POST yang dikirimkan
val content = "Ada alert"
val postData = content.toByteArray(Charsets.UTF_8)
connection.doOutput = true
connection.outputStream.use { os -> os.write(postData) }

val responseCode = connection.responseCode
// Proses responseCode atau response body jika diperlukan

//tutup koneksi
connection.disconnect()
```

Menggunakan string.xml untuk referensi variabel teks

- Buka atau buat file **res/values/strings.xml** jika belum ada.
- Tambahkan string dengan memberikan nama dan nilai untuk teks "Kirim Alert":

```
<resources>
    <string name="app_name">Panic Button</string>
    <string name="button_text_send">Kirim Alert</string>
</resources>
```

Sekarang, di dalam file layout XML Anda, Anda dapat menggantikan teks langsung dengan referensi ke string yang telah Anda tambahkan:

xml

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

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"  
        android:text="@string/button_text_send"  
        android:layout_centerInParent="true"/>  
</RelativeLayout>
```

Dengan menggunakan referensi string ini, Anda dapat dengan mudah mengelola dan memperbarui teks tombol di seluruh aplikasi jika perlu. Ini juga berguna untuk mendukung berbagai bahasa karena nilai string dapat diterjemahkan ke dalam beberapa bahasa yang didukung.

Latihan:

- coba ganti button di layout dengan **ImageButton** dengan gambar tombol berwarna merah.

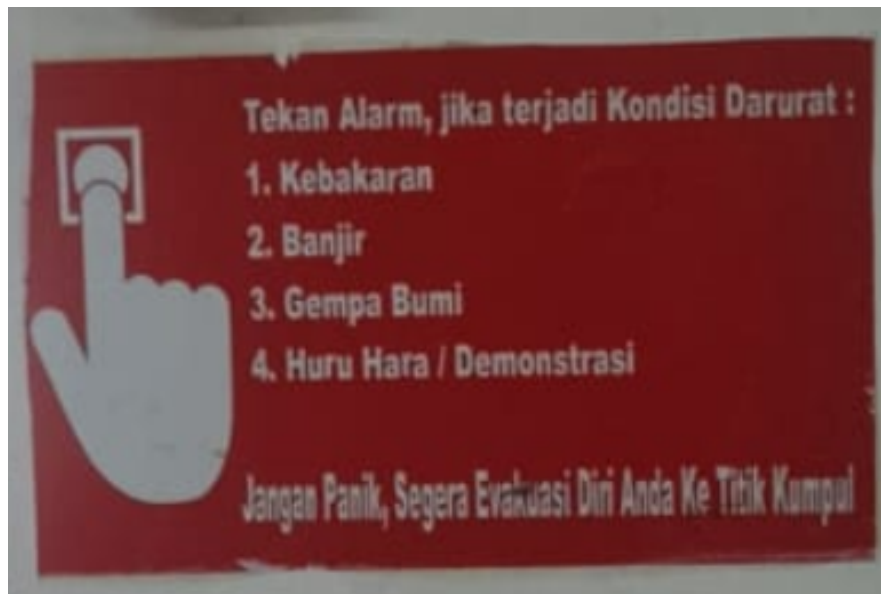
```
<ImageButton  
    android:id="@+id/imageButton"  
    android:layout_width="145dp"  
    android:layout_height="365dp"  
    app:srcCompat="@drawable/red_button" />
```

tambahkan gambar **red_button.jpg** di res/drawable/

B. Sesi Praktikum Kedua

Pada sesi praktikum kedua, aplikasi yang sudah dibuat pada praktikum pertama akan dikembangkan lebih lanjut, terutama menambahkan aspek tampilan seperti layout, navigasi, toast, dll.

Pada panic button fisik tertera pengumuman bahwa tombol panik ditekan jika ada kondisi darurat tertentu seperti gambar di bawah ini.



Gambar petunjuk tombol panik

Pada praktikum ini akan dicoba dibuat activity untuk mengirimkan informasi tambahan seperti pengumuman tersebut.

Pelapor dapat menambahkan jenis laporan kondisi darurat berupa:

1. Kebakaran
2. Banjir
3. Gempa Bumi
4. Huru Hara/Demostrasi
5. Lainnya

Selain itu ditambahkan isian keterangan/catatan tambahan terkait kondisi darurat yang terjadi sehingga lebih jelas pada saat diterima oleh tim respon kondisi darurat.

Kondisi darurat bisa lebih dari satu (check box) karena kejadian darurat bisa lebih dari satu jenis terjadi bersamaan. kemudian tambahkan tulisan seperti yang ada di pengumuman:

**“JANGAN PANIK, SEGERA EVAKUASI
DIRI ANDA KE TITIK KUMPUL”**

Perancangan Tampilan

Sebelum diimplementasikan sebaiknya UI/UX dirancang terlebih dahulu sehingga ada gambaran seperti apa nanti tampilan aplikasi. Tampilan dapat dirancang dengan berbagai cara di antaranya:

- gambar manual di kertas
- drawing tools seperti draw.io, visio, dll
- wireframe dengan sitaks salt <https://plantuml.com/salt>
- Figma <https://www.figma.com/>
- Penpot, Plasmic, Wireflow, dll

Contoh merancang tampilan menggunakan salt wireframe:

Terjadi Kondisi Darurat

☒ Kebakaran

☐ Banjir

☐ Gempa Bumi

☐ Huru hara/Demostrasi

☐ Lainnya

Catatan tambahan ...

Kirim Laporan

**“JANGAN PANIK!, SEGERA EVAKUASI
DIRI ANDA KE TITIK KUMPUL”**

Layout tampilan menggunakan Salt PlantUML:

```

@startsalt
{
  Terjadi Kondisi Darurat
  [X] Kebakaran
  [] Banjir
  [] Gempa Bumi
  [] Huru hara/Demonstrasi
  [] Lainnya
  "Catatan tambahan ...  "

  [Kirim Laporan]
  ==
  <color:red><size:15> **"JANGAN PANIK!, SEGERA EVAKUASI**
  <color:red><size:15> **DIRI ANDA KE TITIK KUMPUL**
}
@endsalt

```

http://plantuml.ubharajaya.ac.id/uml/TL3DliD04BxdAKQFGL8yU2bjwTQ4k6uDfKsaK7AOceLkxMxAxEQW8IH1zENw94wiLndu-7xcOs2cFY05ZuSGVKG0fQOzDWPKwnhZ3QH87M7WPBEkGUajlYAXwtKDCtHxGnUUQtj4c7NMNFBH3HntiU7UGQjiw7mZzNpH7uzmRihY-ny3Z8A3Tuev9aYJ0Fd08FMoK8MEvScpvRFthZSSCevVsQDAGT3CPUVEkXxTt4uZZy-dhKHIVa60X2gckRc2LvTbIGFOiL2LMCex_BwToAK4KgG2LGlbBgK1LJujgVZvzS-ypcchNzA_w0G00

IMPLEMENTASI:

Tambahkan Activity Baru:

- Buka proyek Android Studio Anda.
- Klik kanan pada package proyek Anda di "Project Explorer".
- Pilih "New" -> "Activity" -> "Empty View Activity" (atau pilih jenis activity yang sesuai dengan kebutuhan Anda).
- Isi nama activity misal “**AlertActivity**” dan detail lainnya, kemudian klik "Finish".

Isi layout dari AlertActivity.

```
<?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"
    android:padding="16dp">

    <TextView
        android:id="@+id/textViewPilihan"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Terjadi Kondisi Darurat:"
        android:layout_marginTop="16dp"/>

    <CheckBox
        android:id="@+id/checkboxKebakaran"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Kebakaran"
        android:layout_below="@id/textViewPilihan"
        android:layout_marginBottom="8dp"/>

    <CheckBox
        android:id="@+id/checkboxBanjir"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Banjir"
        android:layout_below="@id/checkboxKebakaran"
        android:layout_marginBottom="8dp"/>

    <CheckBox
        android:id="@+id/checkboxGempaBumi"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Gempa Bumi"
        android:layout_below="@id/checkboxBanjir"
        android:layout_marginBottom="8dp"/>

    <CheckBox
        android:id="@+id/checkboxHuruHara"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Huru-Hara/Demonstrasi"
        android:layout_below="@id/checkboxGempaBumi"
        android:layout_marginBottom="8dp"/>
```

```

<CheckBox
    android:id="@+id/checkboxLainnya"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Lainnya"
    android:layout_below="@id/checkboxHuruHara"
    android:layout_marginBottom="16dp"/>

<EditText
    android:id="@+id/editTextText2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="text"
    android:layout_below="@id/checkboxLainnya"
    android:text="Catatan lainnya" />

<Button
    android:id="@+id/buttonKirimAlert"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Kirim Alert"
    android:layout_below="@id/editTextText2"
    android:layout_marginTop="16dp"/>

<TextView
    android:id="@+id/textViewHatiHati"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="JANGAN PANIK, SEGERA EVAKUASI DIRI ANDA KE TITIK
KUMPUL"
    android:textColor="@color/red"
    android:layout_below="@id/buttonKirimAlert"
    android:layout_marginTop="16dp"/>

</RelativeLayout>

```

Tambahkan referensi warna merah di **colors.xml**

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFF</color>
    <color name="red">#FF0000</color>

```

```
</resources>
```

Latihan:

coba ubah ukuran font menjadi lebih besar untuk teks “Terjadi kondisi darurat.”
dan keterangan dibuat menjadi “bold”
tambahkan tombol tutup dan tombol lihat “jalur evakuasi”

Jawaban:

tambahkan atribut berikut

```
android:textStyle="bold"
```

```
android:textSize="30sp"
```

untuk menambahkan tombol, tambahkan script berikut:

```
<Button
    android:id="@+id/buttonJalurEvakuasi"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Lihat Jalur Evakuasi"
    android:layout_below="@id/textViewHatiHati"
    android:layout_marginTop="16dp"/>

<Button
    android:id="@+id/buttonTutup"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Tutup/Batal"
    android:layout_below="@id/buttonJalurEvakuasi"
    android:layout_marginTop="26dp"/>
```

Memanggil **AlertActivity** dari **MainActivity**

Buka file MainActivity.kt

Tambahkan kode berikut untuk membuat objek **Intent** dan memanggil activity baru menggunakan **intent**.

```
import android.content.Intent
```

Contoh:

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_main)  
  
    val button = findViewById<Button>(R.id.button)  
    val intent = Intent(this, AlertActivity::class.java)  
    button.setOnClickListener {  
        startActivity(intent)  
    }  
}
```

Sekarang, ketika Anda menjalankan aplikasi, activity **AlertActivity** seharusnya dapat diakses dari **MainActivity** ketika tombol ditekan.

Selanjutnya:

- Tambahkan listener ketika tombol **tutup/close** ditekan dan **AlertActivity** distop/finish.
- Tambahkan listener ketika tombol kirim alert ditekan,
- pindahkan script **SendPostRequest** yang tadinya ada di **mainactivity.kt** ke **AlertActivity.kt**

tambahkan listener untuk tombol tutup dan tutup activity dalam callback **onCreate**

```
val closeButton: Button = findViewById(R.id.buttonTutup)  
closeButton.setOnClickListener {  
    // Memanggil metode finish() untuk menutup activity saat tombol  
    diklik
```

```
        finish()
    }
}
```

Memindahkan SendPostRequest menjadi fungsi “KirimAlert”

```
package id.ac.ubharajaya.panicbutton

import android.os.AsyncTask
import android.os.Bundle
import android.widget.Button
import androidx.appcompat.app.AppCompatActivity
import java.net.HttpURLConnection
import java.net.URL

class AlertActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_alert)

        // Memanggil metode finish() untuk menutup activity saat tombol diklik
        val closeButton: Button = findViewById(R.id.buttonTutup)
        closeButton.setOnClickListener {
            finish()
        }

        val buttonKirimAlert = findViewById<Button>(R.id.buttonKirimAlert)
        buttonKirimAlert.setOnClickListener {
            // Kode yang akan dijalankan saat tombol diklik
            SendAlert().execute()
        }
    }

    private inner class SendAlert : AsyncTask<Void, Void, Void>() {
        override fun doInBackground(vararg params: Void?): Void? {
            try {
                val url = URL("https://ntfy.ubharajaya.ac.id/panic-button")
                val connection = url.openConnection() as HttpURLConnection
                connection.requestMethod = "POST"

                // ... (kode yang sama dengan yang Anda punya)

                // Proses responseCode atau response body jika diperlukan
            } catch (e: Exception) {
                e.printStackTrace()
            }
        }
    }
}
```

```

// Atur header request dengan informasi yang diinginkan
        connection.setRequestProperty("Content-Type", "text/plain")
        connection.setRequestProperty("Title", "Alert")
        connection.setRequestProperty("Priority", "urgent")
        connection.setRequestProperty("Tags", "alert
warning,rotating_light")

// Tambahkan konten ke permintaan POST yang dikirimkan
        val content = "Ada alert"
        val postData = content.toByteArray(Charsets.UTF_8)
        connection.doOutput = true
        connection.outputStream.use { os -> os.write(postData) }
        val responseCode = connection.responseCode

// Proses responseCode atau response body jika diperlukan

//tutup koneksi
        connection.disconnect()

    } catch (e: Exception) {
        e.printStackTrace()
    }
    return null
}
}
}

```

Mengambil value isian dari user

Dapatkan nilai dari **EditText** dan **Checkbox**:

Import widget edit text dan Checkbox

```

import android.widget.EditText
import android.widget.CheckBox

```

Ambil value catatan lainnya dan checkbox

```

//Ambil catatan lainnya
val catatanLainnyaEditText = findViewById<EditText>(R.id.editTextText2)
val catatanLainnya = catatanLainnyaEditText.text.toString()

```

```
//ambil isian checkbox
val checkboxes = arrayOf(
    findViewById<CheckBox>(R.id.checkboxKebakaran),
    findViewById<CheckBox>(R.id.checkboxBanjir),
    findViewById<CheckBox>(R.id.checkboxGempaBumi),
    findViewById<CheckBox>(R.id.checkboxHuruHara),
    findViewById<CheckBox>(R.id.checkboxLainnya)
)

val selectedCheckboxValues = checkboxes
    .filter { it.isChecked }
    .map { it.text.toString() }
    .joinToString(" dan ")

// Tambahkan konten ke permintaan POST yang dikirimkan

val content = "Terjadi kondisi darurat berikut: $selectedCheckboxValues
catatan: $catatanLainnya"
```

Tambahkan nilai tersebut ke header permintaan POST.

Hasil akhir:

```
package id.ac.ubharajaya.panicbutton
import android.widget.EditText
import android.widget.CheckBox

import android.os.AsyncTask
import android.os.Bundle
import android.widget.Button
import androidx.appcompat.app.AppCompatActivity
import java.net.HttpURLConnection
import java.net.URL

class AlertActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_alert2)

        val buttonKirimAlert = findViewById<Button>(R.id.buttonKirimAlert)

        buttonKirimAlert.setOnClickListener {
            // Kode yang akan dijalankan saat tombol diklik
            KirimAlert().execute()
        }
    }
}
```

```

private inner class KirimAlert : AsyncTask<Void, Void, Void>() {
    override fun doInBackground(vararg params: Void?): Void? {
        try {
            val url = URL("https://ntfy.ubharajaya.ac.id/panic-button")
            val connection = url.openConnection() as HttpURLConnection
            connection.requestMethod = "POST"

            // ... (kode yang sama dengan yang Anda punya)

            // Proses responseCode atau response body jika diperlukan

// Atur header request dengan informasi yang diinginkan
            connection.setRequestProperty("Content-Type", "text/plain")
            connection.setRequestProperty("Title", "Alert")
            connection.setRequestProperty("Priority", "urgent")
            connection.setRequestProperty("Tags", "alert
warning,rotating_light")

            val catatanLainnyaEditText =
findViewById<EditText>(R.id.editTextText2)
            val catatanLainnya = catatanLainnyaEditText.text.toString()

            val checkboxes = arrayOf(
                findViewById<CheckBox>(R.id.checkboxKebakaran),
                findViewById<CheckBox>(R.id.checkboxBanjir),
                findViewById<CheckBox>(R.id.checkboxGempaBumi),
                findViewById<CheckBox>(R.id.checkboxHuruHara),
                findViewById<CheckBox>(R.id.checkboxLainnya)
            )

            val selectedCheckboxValues = checkboxes
                .filter { it.isChecked }
                .map { it.text.toString() }
                .joinToString(", ")

// Tambahkan konten ke permintaan POST yang dikirimkan

            val content = "$catatanLainnya - Terjadi kondisi darurat:
$selectedCheckboxValues"

            val postData = content.toByteArray(Charsets.UTF_8)
            connection.doOutput = true
            connection.outputStream.use { os -> os.write(postData) }
            val responseCode = connection.responseCode

```

```
// Proses responseCode atau response body jika diperlukan

//tutup koneksi
        connection.disconnect()

        } catch (e: Exception) {
            e.printStackTrace()
        }
        return null
    }
}
}
```

final code untuk AlertActivity.kt

```
package id.ac.ubharajaya.panicbutton

import android.widget.EditText
import android.widget.CheckBox

import android.os.AsyncTask
import android.os.Bundle
import android.widget.Button
import androidx.appcompat.app.AppCompatActivity
import java.net.HttpURLConnection
import java.net.URL

class AlertActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_alert2)

        val closeButton: Button = findViewById(R.id.buttonTutup)
        closeButton.setOnClickListener {
            // Memanggil metode finish() untuk menutup activity saat tombol
            diklik
            finish()
        }
    }
}
```

```

        val buttonKirimAlert = findViewById<Button>(R.id.buttonKirimAlert)
        buttonKirimAlert.setOnClickListener {
            // Kode yang akan dijalankan saat tombol diklik
            SendAlert().execute()
        }
    }

    private inner class SendAlert : AsyncTask<Void, Void, Void>() {
        override fun doInBackground(vararg params: Void?): Void? {
            try {
                val url = URL("https://ntfy.ubharajaya.ac.id/panic-button")
                val connection = url.openConnection() as HttpURLConnection
                connection.requestMethod = "POST"

                // ... (kode yang sama dengan yang Anda punya)

                // Proses responseCode atau response body jika diperlukan

                // Atur header request dengan informasi yang diinginkan
                connection.setRequestProperty("Content-Type", "text/plain")
                connection.setRequestProperty("Title", "Alert")
                connection.setRequestProperty("Priority", "urgent")
                connection.setRequestProperty("Tags", "alert warning,
rotating_light")

                val catatanLainnyaEditText =
                findViewById<EditText>(R.id.editTextText2)
                val catatanLainnya = catatanLainnyaEditText.text.toString()

                val checkboxes = arrayOf(
                    findViewById<CheckBox>(R.id.checkboxKebakaran),
                    findViewById<CheckBox>(R.id.checkboxBanjir),
                    findViewById<CheckBox>(R.id.checkboxGempaBumi),
                    findViewById<CheckBox>(R.id.checkboxHuruHara),
                    findViewById<CheckBox>(R.id.checkboxLainnya)
                )

                val selectedCheckboxValues = checkboxes
                    .filter { it.isChecked }
                    .map { it.text.toString() }
                    .joinToString(", ")

                // Tambahkan konten ke permintaan POST yang dikirimkan
                //          val content = catatanLainnya

```

```

        val content = " Terjadi: $selectedCheckboxValues - Catatan:
$catatanLainnya"

        val postData = content.toByteArray(Charsets.UTF_8)
        connection.doOutput = true
        connection.outputStream.use { os -> os.write(postData) }
        val responseCode = connection.responseCode

// Proses responseCode atau response body jika diperlukan

//tutup koneksi
        connection.disconnect()

    } catch (e: Exception) {
        e.printStackTrace()
    }
    return null
}
}
}

```

LATIHAN (PR):

- Ubah tag sesuai kiriman user, misal kalau kebakaran akan dimunculkan emoji api, banjir air, dll.. <https://docs.ntfy.sh/emojis/>
 - Kebakaran 
 - Banjir 
 - Tsunami 
 - Gunung Meletus 
 - Gempa Bumi 
 - Huru Hara 
 - Binatang Buas 
 - Radiasi Nuklir 
 - Biohazard 
- Latihan lanjutan, coba buat satu activity lagi yang memunculkan “**jalur evakuasi**” dengan nama activity JalurEvakuasiActivity menuju lokasi titik kumpul dan ada tombol close seperti gambar di bawah ini:



[TUTUP]

Activity akan dimunculkan ketika tombol di bawah informasi **“JANGAN PANIK, SEGERA EVAKUASI DIRI ANDA KE TITIK KUMPUL”** atau di tombol lain yang sesuai. Ketika activity ditutup maka akan kembali ke activity sebelumnya.

```
<Button
    android:id="@+id/buttonJalurEvakuasi"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Lihat Jalur Evakuasi"
    android:layout_below="@id/textViewHatiHati"
    android:layout_marginTop="16dp"/>
```

tambahkan juga button jalur evakuasi pada mainActivity

C. Sesi Praktikum Ketiga

Pada sesi ketiga kita akan menggunakan fitur menu dan menyimpan data dan konfigurasi di lokal Android dan menambahkan konfirmasi jika pesan terkirim.

Menambahkan konfirmasi jika pesan terkirim.

1. Membuat PopUpMenu
2. Menyimpan Data dan Konfigurasi di SharedPreferences

Setting yang perlu disimpan

- NTFY URL <https://ntfy.ubharajaya.ac.id/panic-button> atau <https://ntfy.sh/ubahara-panic-button>
- selanjutnya coba pisahkan server dan topik
 - server: <https://ntfy.sh>
 - topic: [ubahara-panic-button](https://ntfy.sh/ubahara-panic-button)

Konfigurasi dalam pengembangan perangkat lunak Android dapat disimpan di berbagai tempat, tergantung pada jenis data yang disimpan.:

di bawah folder **res/menu/** buat file **popup_menu.xml**

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/menuSettings"
        android:title="Settings"/>
    <item
        android:id="@+id/menuJalurEvakuasi"
        android:title="Jalur Evakuasi"/>
    <!-- Tambahkan lebih banyak item menu jika diperlukan -->
</menu>
```

Latihan:

tambahkan menu “Bantuan/Help” dan “About”

Tambahkan sub menu, pilihan gedung untuk jalur evakuasi

- Gedung Said Sukanto
- Gedung Yasin
- Gedung Sumarecon
- Gedung Tanoto

Jawaban:

```
<item
    android:id="@+id/menuHelp"
    android:title="Help"/>
<item
    android:id="@+id/menuAbout"
    android:title="Help"/>
```

Isi MainActivity.kt

```
package id.ac.ubharajaya.panicbutton
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.TextView
import android.view.View
import android.widget.PopupMenu
import android.view.MenuItem

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val button = findViewById<Button>(R.id.button)
        val intent = Intent(this, AlertActivity::class.java)
        button.setOnClickListener {
            startActivity(intent)
        }

        val buttonEvacuationActivity =
            findViewById<Button>(R.id.jalurEvakuasi)
        buttonEvacuationActivity.setOnClickListener {
            openJalurEvakuasi()
        }
    }
}
```

```

// Mendapatkan referensi ke TextView yang digunakan sebagai menu
val menuTextView: TextView = findViewById(R.id.menuTextView)

// Menambahkan pendengar klik untuk TextView menu
menuTextView.setOnClickListener {
    // Memanggil metode untuk menampilkan menu popup
    showPopupMenu(it)
}
}

private fun showPopupMenu(view: View) {
    // Membuat instance dari PopupMenu
    val popupMenu = PopupMenu(this, view)

    // Menginflasi menu dari file XML
    popupMenu.menuInflater.inflate(R.menu.popup_menu, popupMenu.menu)

    // Menambahkan pendengar klik untuk item menu
    popupMenu.setOnMenuItemClickListener { item ->
        // Menghandle klik item menu
        when (item.itemId) {
            R.id.menuSettings -> {
                // Handle klik Menu Item 1
                openSettings()
                true
            }
            R.id.menuJalurEvakuasi -> {
                // Handle klik Menu Item 2
                openJalurEvakuasi()
                true
            }
            R.id.menuHelp -> {
                // Handle klik Menu Item 2
                openHelp()
                true
            }
            // Tambahkan lebih banyak kasus untuk item menu tambahan jika
            diperlukan
            else -> false
        }
    }

    // Menampilkan menu popup
    popupMenu.show()
}

```

```

private fun openSettings() {
    val intent = Intent(this, SettingsActivity::class.java)
    startActivity(intent)
}

private fun openJalurEvakuasi() {
    val intentEvac = Intent(this, EvacuationActivity::class.java)
    startActivity(intentEvac)
}

private fun openHelp() {
    val intentHelp = Intent(this, HelpActivity::class.java)
    startActivity(intentHelp)
}
}

```

Latihan:

Buat ActivityHelp yang dipanggil ketika user klik help.

Buat SettingActivity

activity_setting.xml

```

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

    <TextView
        android:id="@+id/textViewUrl"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="URL NTFY:"
        android:layout_marginTop="16dp"
        android:layout_marginStart="16dp"/>

    <EditText
        android:id="@+id/urlEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/textViewUrl"
        android:layout_marginTop="8dp"

```

```

        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:hint="Masukkan URL"/>

<Button
    android:id="@+id/saveButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Simpan"
    android:layout_below="@id/urlEditText"
    android:layout_marginTop="16dp"
    android:layout_marginStart="16dp"
    android:onClick="saveUrl"/>

</RelativeLayout>

```

SettingActiviyt.kt

```

package id.ac.ubharajaya.panicbutton
import android.content.Context
import android.content.SharedPreferences
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import android.widget.Button
import android.widget.EditText

class SettingsActivity : AppCompatActivity() {
    private lateinit var sharedPreferences: SharedPreferences
    private lateinit var urlEditText: EditText
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_settings)

        // Inisialisasi SharedPreferences
        sharedPreferences = getSharedPreferences("MyAppPreferences",
Context.MODE_PRIVATE)

        // Tampilkan nilai URL saat ini (jika ada)
        val currentUrl = sharedPreferences.getString("urlKey", "")
        // Inisialisasi urlEditText
        urlEditText = findViewById(R.id.urlEditText)

        urlEditText.setText(currentUrl)
    }
}

```

```

        // Jika nilai URL kosong, set nilai awal di sini
        if (currentUrl.isNullOrEmpty()) {

            urlEditText.setText("https://ntfy.ubharajaya.ac.ac.id/panic-button")
        }

        val saveButton = findViewById<Button>(R.id.saveButton)
        saveButton.setOnClickListener {
            saveUrl()
        }
    }

    private fun saveUrl() {
        val url = urlEditText.text.toString().trim()

        // Validasi URL (Tambahkan validasi sesuai kebutuhan aplikasi Anda)

        // Simpan URL ke SharedPreferences
        val editor = sharedPreferences.edit()
        editor.putString("urlKey", url)
        editor.apply()
        finish()
    }
}

```

Ubah AlertActivity.kt untuk mengambil preference “urlKey”

```

private lateinit var sharedPreferences: SharedPreferences

val url = URL(sharedPreferences.getString("urlKey",
"https://ntfy.ubharajaya.ac.id/panic-button") ?: "")

val connection = url.openConnection() as HttpURLConnection
connection.requestMethod = "POST"

```

hasil akhir AlertActivity.kt

```

package id.ac.ubharajaya.panicbutton

import android.content.Context
import android.content.SharedPreferences

import android.widget.EditText

```

```

import android.widget.CheckBox

import android.os.AsyncTask
import android.os.Bundle
import android.widget.Button
import androidx.appcompat.app.AppCompatActivity
import java.net.HttpURLConnection
import java.net.URL

class AlertActivity : AppCompatActivity() {
    private lateinit var sharedPreferences: SharedPreferences

    override fun onCreate(savedInstanceState: Bundle?) {1
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_alert2)

        // Inisialisasi SharedPreferences
        sharedPreferences = getSharedPreferences("MyAppPreferences",
Context.MODE_PRIVATE)

        val closeButton: Button = findViewById(R.id.buttonTutup)
        closeButton.setOnClickListener {
            // Memanggil metode finish() untuk menutup activity saat tombol
diklik
            finish()
        }

        val buttonKirimAlert = findViewById<Button>(R.id.buttonKirimAlert)
        buttonKirimAlert.setOnClickListener {
            // Kode yang akan dijalankan saat tombol diklik
            SendAlert().execute()
        }
    }

    private inner class SendAlert : AsyncTask<Void, Void, Void>() {

        override fun doInBackground(vararg params: Void?): Void? {
            try {
                // val url = URL("https://ntfy.ubharajaya.ac.id/panic-button")
                // val url = URL(sharedPreferences.getString("urlKey",
"https://ntfy.ubharajaya.ac.id/panic-button") ?: "")
                val url = URL(sharedPreferences.getString("urlKey",
"https://ntfy.ubharajaya.ac.id/panic-button") ?:
"https://ntfy.ubharajaya.ac.id/panic-button")

                val connection = url.openConnection() as HttpURLConnection

```

```

        connection.requestMethod = "POST"

        // ... (kode yang sama dengan yang Anda punya)

        // Proses responseCode atau response body jika diperlukan

// Atur header request dengan informasi yang diinginkan
        connection.setRequestProperty("Content-Type", "text/plain")
        connection.setRequestProperty("Title", "Alert")
        connection.setRequestProperty("Priority", "urgent")
        connection.setRequestProperty("Tags", "alert warning,
rotating_light")

        val catatanLainnyaEditText =
findViewById<EditText>(R.id.editTextText2)
        val catatanLainnya = catatanLainnyaEditText.text.toString()

        val checkboxes = arrayOf(
            findViewById<CheckBox>(R.id.checkboxKebakaran),
            findViewById<CheckBox>(R.id.checkboxBanjir),
            findViewById<CheckBox>(R.id.checkboxGempaBumi),
            findViewById<CheckBox>(R.id.checkboxHuruHara),
            findViewById<CheckBox>(R.id.checkboxLainnya)
        )

        val selectedCheckboxValues = checkboxes
            .filter { it.isChecked }
            .map { it.text.toString() }
            .joinToString(", ")

// Tambahkan konten ke permintaan POST yang dikirimkan
//         val content = catatanLainnya

        val content = " Terjadi: $selectedCheckboxValues - Catatan:
$catatanLainnya"

        val postData = content.toByteArray(Charsets.UTF_8)
        connection.doOutput = true
        connection.outputStream.use { os -> os.write(postData) }
        val responseCode = connection.responseCode

// Proses responseCode atau response body jika diperlukan

//tutup koneksi

```

```
        connection.disconnect()

    } catch (e: Exception) {
        e.printStackTrace()
    }
    return null
}
}
```

Latihan/Tugas:

Ubah setting URL misal <https://ntfy.ubharajaya.ac.id/panic-button> dipisah menjadi

- **Server NTFY**, contoh “<https://ntfy.ubharajaya.ac.id>” atau <https://ntfy.sh>
- **Topic**, contoh “**darurat**” atau “**panic-button**”

D. Tugas Kelompok

Untuk tugas kelompok, aplikasi mobile yang dibuat pada saat praktikum dapat dijadikan **“starter”** untuk dikembangkan lebih lanjut. Jika tim sudah memiliki ide aplikasi bisa juga digunakan sebagai tugas kelompok.

Maksimum 2 atau 3 mhs seperti yang sudah dibagi.

Beberapa hal yang bisa ditambahkan:

- Tambahkan **“splash”/animasi** aplikasi dengan logo Ubhara Jaya dan ganti logo aplikasi.
- Ambil koordinat lokasi pelapor dengan fitur GPS dan kirimkan dalam alert.
- Munculkan lokasi pelapor dan titik kumpul di peta sebenarnya (bukan gambar)
- Setelah data sukses dikirimkan maka akan dimunculkan konfirmasi ke user dalam bentuk **“toast”** atau **“snackbar”**. “Pesan terkirim!” atau “Gagal Mengirimkan Pesan”.
- Tambahkan informasi foto/gambar pada laporan NTFY
- Buat database lokal untuk menyimpan hasil pengiriman.
- Tambahkan webview, fragment dll..
- dll

Tugas kelompok dikumpulkan **satu pekan sebelum UAS**, pengumpulan berupa repository Github, Gitlab, Bitbucket, codeberg ke dosen pengampu dan slide/paparan aplikasi.