

UTS PEMOGRAMAN MOBILE 1

Dosen Pengampu: Nova Agustina, ST., M.Kom.



Disusun Oleh :

Nama : Fahmi Fauziah Nur Fadillah

NPM : 23552011314

Kelas : TIF RP 23A CNS

DEPARTEMEN TEKNIK INFORMATIKA

FAKULTAS INDUSTRI KREATIF

UNIVERSITAS TEKNOLOGI BANDUNG

2025

ESSAY

1. Apa fungsi findViewById?

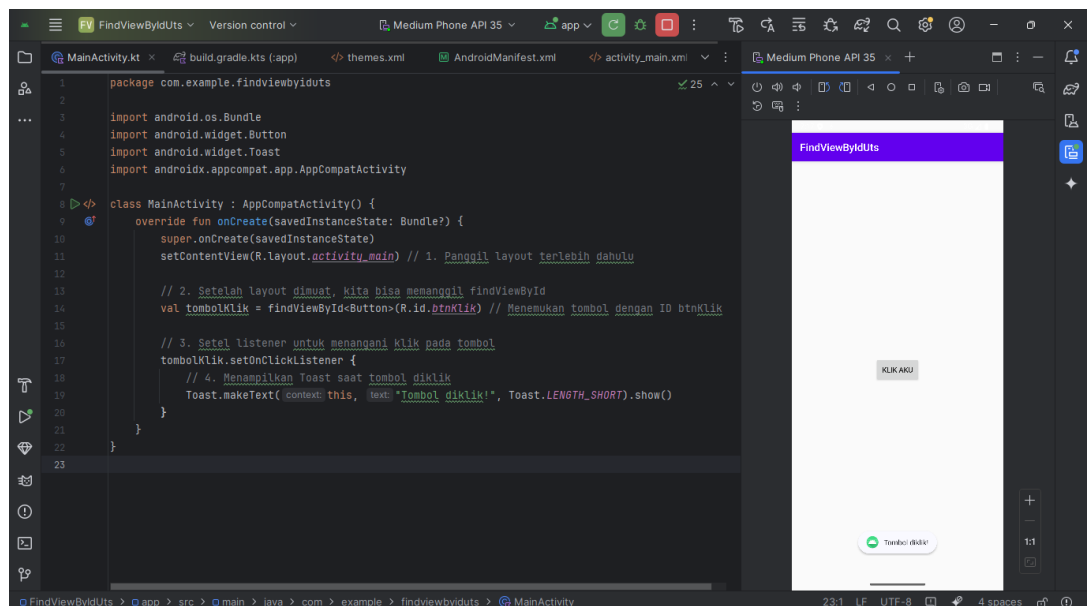
Jawaban:

findViewById adalah sebuah fungsi di Android yang digunakan untuk menghubungkan elemen tampilan (view) yang didefinisikan di file XML dengan kode program Kotlin. Fungsi ini mencari elemen berdasarkan ID-nya dan mengembalikan objek View yang sesuai, sehingga bisa dimanipulasi melalui kode.

2. Apa syarat pemanggilan method findViewById? Buat contohnya dan screenshot source code nya!

Jawaban:

- Harus ada layout yang sudah di-setContentView: findViewById hanya bisa dipanggil setelah layout XML diterapkan menggunakan setContentView. Artinya, setContentView(R.layout.activity_main) harus dipanggil terlebih dahulu untuk memastikan layout sudah tersedia sebelum kamu mencoba mengakses elemen-elemen di dalamnya.
- Harus sesuai dengan ID komponen yang ada di layout: ID yang digunakan di findViewById harus sesuai dengan ID yang ada di file XML. Misalnya, di XML saya punya <Button android:id="@+id/btnKlik" />, maka yang harus dipanggil adalah findViewById<Button>(R.id.btnKlik) di kode.

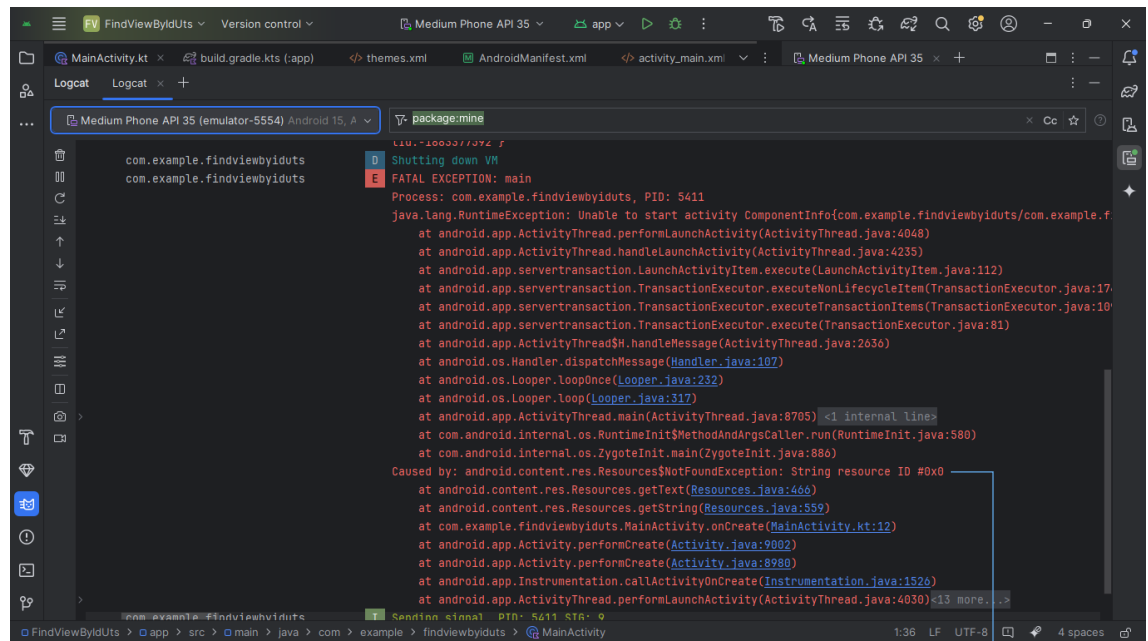


3. Error apa yang terjadi jika file kotlin salah menginisialisasi findViewById atau objek pada xml belum diinisialisasi?

Jawaban:

Jika tidak menginisialisasi findViewById dengan benar atau objek dalam XML belum terhubung, beberapa error yang bisa muncul adalah:

- **NullPointerException**: Hal ini dapat terjadi ketika akan mencoba untuk mengakses elemen UI yang belum diinisialisasi dengan benar. Misalnya, jika ada salah menetik ID atau ID tersebut tidak ada dalam layout XML, maka objek yang dihasilkan akan null, dan jika dicoba untuk mengakses atau memodifikasi objek tersebut, maka akan muncul **NullPointerException**.
 - **Resources.NotFoundException**: Kesalahan ini terjadi ketika akan mencoba mengakses ID yang tidak ditemukan di dalam layout XML. Jika ID yang dipanggil tidak ada, maka Android akan melemparkan **Resources.NotFoundException**.
4. Buat sebuah contoh program untuk menampilkan pesan error **Resources.NotFoundException**! Screenshot logcat-nya!



Resources.NotFoundException
yang ada di logcat

STUDI KASUS

1. Buatlah sebuah program sederhana yang terdiri dari 4 Activity menggunakan Android Native (Java + XML) yang terdiri dari:
 - a SplashScreen Activity
 - b Login Activity
 - c Register Activity

d List Chating

2. Ketentuan: Silahkan membuat splashscreen dengan baik.
3. Pada Register Activity, minimal terdapat objek: TextView, EditText, Button, ImageView!
4. Tampilkan event Log, Toast dan Toast pada saat Button Register di klik.
5. Pada List Chating terdapat data yang ditampilkan dalam listview
6. Upload project di Github.
7. **Jelaskan fungsi setiap baris source code pada file kotlin dan submit dalam bentuk pdf pada Elearning**

JAWABAN:

Link Github: https://github.com/Fahmifzh/UTS_PEMROGRAMAN_MOBILE1

- Judul Aplikasi: LaporDesaku

Aplikasi LaporDesaku adalah aplikasi yang memberikan kemudahan bagi warga desa untuk menyampaikan pengaduan dan mengakses informasi desa secara digital. Aplikasi ini dirancang sederhana agar mudah digunakan oleh semua warga, bahkan yang belum terbiasa dengan teknologi.

Tujuan Aplikasi LaporDesaku:

- Fungsi Utama / Apa Saja yang Bisa Dilakukan dalam Aplikasi:

1. Registrasi dan Login:

Warga bisa mendaftarkan akun dan masuk untuk mulai menggunakan aplikasi.

2. Tambah Pengaduan:

Warga dapat mengisi formulir laporan (judul, kategori, deskripsi) dan menyimpannya.

3. Lihat Laporan Saya:

Warga bisa melihat daftar laporan yang pernah mereka kirimkan.

4. Lihat List Pengaduan Warga Lain:

Semua warga bisa melihat laporan yang dikirimkan warga lain (transparansi), tapi tidak bisa mengedit.

5. Chat Warga:

Warga dapat berdiskusi atau berbincang seputar kehidupan desa melalui simulasi percakapan.

6. Informasi Tentang Aplikasi:

Penjelasan mengenai tujuan aplikasi, kontak layanan desa, dan info pengembang (opsional).

PENJELASAN FILE KOTLIN

1. SplashScreen	package com.example.lapordesaku // Mendefinisikan nama paket aplikasi
-----------------	---

	<pre> import android.content.Intent // Mengimpor kelas Intent untuk berpindah antar aktivitas import android.os.Bundle // Mengimpor kelas Bundle untuk menyimpan dan memulihkan data instance aktivitas import android.os.Handler // Mengimpor kelas Handler untuk menjalankan kode pada thread utama dengan penundaan import androidx.appcompat.app.AppCompatActivity // Mengimpor kelas AppCompatActivity sebagai kelas dasar untuk aktivitas yang menggunakan fitur tema dan toolbar AppCompatActivity class SplashScreenActivity : AppCompatActivity() { // Mendefinisikan kelas SplashScreenActivity yang merupakan turunan dari AppCompatActivity private val splashDelay: Long = 3000 // Mendeklarasikan konstanta untuk durasi tampilan splash screen dalam milidetik (3 detik) override fun onCreate(savedInstanceState: Bundle?) { // Metode yang dipanggil saat aktivitas pertama kali dibuat super.onCreate(savedInstanceState) // Memanggil implementasi onCreate dari kelas induk setContentView(R.layout.activity_splash_screen) // Menghubungkan layout XML (activity_splash_screen.xml) dengan aktivitas ini Handler().postDelayed({ // Membuat instance Handler dan memposting Runnable untuk dijalankan setelah penundaan val intent = Intent(this, LoginActivity::class.java) // Membuat Intent untuk berpindah ke aktivitas LoginActivity startActivity(intent) // Memulai aktivitas LoginActivity finish() // Menutup aktivitas SplashScreenActivity agar pengguna tidak bisa kembali ke splash screen dengan tombol back }, splashDelay) // Menentukan durasi penundaan (dalam milidetik) sebelum Runnable dijalankan } } } </pre>
2. login	<pre> package com.example.lapordesaku // Mendefinisikan nama paket aplikasi import android.content.Intent // Mengimpor kelas Intent untuk berpindah antar aktivitas import android.os.Bundle // Mengimpor kelas Bundle untuk menyimpan dan memulihkan data instance aktivitas </pre>

	<pre> import android.widget.Button // Mengimpor kelas Button untuk elemen tombol interaktif import android.widget.EditText // Mengimpor kelas EditText untuk elemen input teks import android.widget.TextView // Mengimpor kelas TextView untuk menampilkan teks import android.widget.Toast // Mengimpor kelas Toast untuk menampilkan pesan singkat kepada pengguna import androidx.appcompat.app.AppCompatActivity // Mengimpor kelas AppCompatActivity sebagai kelas dasar untuk aktivitas yang menggunakan fitur tema dan toolbar AppCompatActivity class LoginActivity : AppCompatActivity() { // Mendefinisikan kelas LoginActivity yang merupakan turunan dari AppCompatActivity private lateinit var emailEditText: EditText // Mendeklarasikan variabel untuk EditText tempat pengguna memasukkan email private lateinit var passwordEditText: EditText // Mendeklarasikan variabel untuk EditText tempat pengguna memasukkan password private lateinit var loginButton: Button // Mendeklarasikan variabel untuk Button yang digunakan untuk melakukan login private lateinit var registerTextView: TextView // Mendeklarasikan variabel untuk TextView yang dapat diklik untuk menuju halaman registrasi override fun onCreate(savedInstanceState: Bundle?) { // Metode yang dipanggil saat aktivitas pertama kali dibuat super.onCreate(savedInstanceState) // Memanggil implementasi onCreate dari kelas induk setContentView(R.layout.activity_login) // Menghubungkan layout XML (activity_login.xml) dengan aktivitas ini emailEditText = findViewById(R.id.emailEditText) // Mendapatkan referensi ke EditText untuk email dari layout berdasarkan ID passwordEditText = findViewById(R.id.passwordEditText) // Mendapatkan referensi ke EditText untuk password dari layout berdasarkan ID loginButton = findViewById(R.id.loginButton) // Mendapatkan referensi ke Button untuk login dari layout berdasarkan ID registerTextView = findViewById(R.id.registerTextView) // Mendapatkan </pre>
--	---

	<p>referensi ke TextView untuk registrasi dari layout berdasarkan ID</p> <pre> loginButton.setOnClickListener { // Menetapkan OnClickListener untuk tombol loginButton val email = emailEditText.text.toString() // Mendapatkan teks email yang dimasukkan pengguna val password = passwordEditText.text.toString() // Mendapatkan teks password yang dimasukkan pengguna // Untuk sementara, kita simulasikan login berhasil jika email dan password tidak kosong if (email.isNotEmpty() && password.isNotEmpty()) { // Memeriksa apakah email dan password tidak kosong // Simpan sesi login (implementasi sederhana, bisa menggunakan SharedPreferences) // Misalnya: saveLoginSession(email) val intent = Intent(this, HomeActivity::class.java) // Membuat Intent untuk berpindah ke aktivitas HomeActivity startActivity(intent) // Memulai aktivitas HomeActivity finish() // Menutup aktivitas LoginActivity agar pengguna tidak bisa kembali ke halaman login dengan tombol back tanpa logout } else { // Jika email atau password kosong Toast.makeText(this, "Email dan password tidak boleh kosong", Toast.LENGTH_SHORT).show() // Menampilkan pesan Toast singkat kepada pengguna } } registerTextView.setOnClickListener { // Menetapkan OnClickListener untuk TextView registerTextView val intent = Intent(this, RegisterActivity::class.java) // Membuat Intent untuk berpindah ke aktivitas RegisterActivity startActivity(intent) // Memulai aktivitas RegisterActivity } } // Fungsi simulasi penyimpanan sesi login (nanti akan diimplementasikan dengan lebih baik) private fun saveLoginSession(email: String) { // Fungsi pribadi untuk menyimpan sesi login (sementara) // Implementasi penyimpanan sesi seperti menggunakan SharedPreferences // Contoh sederhana: </pre>
--	---

	<pre> val sharedPreferences = getSharedPreferences("login_session", MODE_PRIVATE) // Mendapatkan atau membuat SharedPreferences dengan nama "login_session" yang hanya dapat diakses oleh aplikasi ini val editor = sharedPreferences.edit() // Membuat editor untuk memodifikasi data dalam SharedPreferences editor.putString("user_email", email) // Menyimpan email pengguna dengan kunci "user_email" editor.apply() // Menerapkan perubahan yang telah dibuat pada SharedPreferences secara asynchronous } } </pre>
3. register	<pre> package com.example.lapordesaku // Mendefinisikan nama paket aplikasi import android.content.Intent // Mengimpor kelas Intent untuk berpindah antar aktivitas import android.os.Bundle // Mengimpor kelas Bundle untuk menyimpan dan memulihkan data instance aktivitas import android.widget.Button // Mengimpor kelas Button untuk elemen tombol interaktif import android.widget.EditText // Mengimpor kelas EditText untuk elemen input teks import android.widget.Toast // Mengimpor kelas Toast untuk menampilkan pesan singkat kepada pengguna import androidx.appcompat.app.AppCompatActivity // Mengimpor kelas AppCompatActivity sebagai kelas dasar untuk aktivitas yang menggunakan fitur tema dan toolbar AppCompatActivity class RegisterActivity : AppCompatActivity() { // Mendefinisikan kelas RegisterActivity yang merupakan turunan dari AppCompatActivity private lateinit var namaEditText: EditText // Mendeklarasikan variabel untuk EditText tempat pengguna memasukkan nama lengkap private lateinit var emailEditText: EditText // Mendeklarasikan variabel untuk EditText tempat pengguna memasukkan alamat email private lateinit var passwordEditText: EditText // Mendeklarasikan variabel untuk EditText tempat pengguna memasukkan kata sandi private lateinit var nomorHpEditText: EditText // Mendeklarasikan variabel untuk EditText tempat pengguna memasukkan nomor telepon private lateinit var alamatEditText: EditText // Mendeklarasikan variabel untuk EditText tempat pengguna memasukkan alamat </pre>

	<pre> private lateinit var registerButton: Button // Mendeklarasikan variabel untuk Button yang digunakan untuk melakukan pendaftaran override fun onCreate(savedInstanceState: Bundle?) { // Metode yang dipanggil saat aktivitas pertama kali dibuat super.onCreate(savedInstanceState) // Memanggil implementasi onCreate dari kelas induk setContentView(R.layout.activity_register) // Menghubungkan layout XML (activity_register.xml) dengan aktivitas ini namaEditText = findViewById(R.id.namaEditText) // Mendapatkan referensi ke EditText untuk nama dari layout berdasarkan ID emailEditText = findViewById(R.id.emailEditText) // Mendapatkan referensi ke EditText untuk email dari layout berdasarkan ID passwordEditText = findViewById(R.id.passwordEditText) // Mendapatkan referensi ke EditText untuk password dari layout berdasarkan ID nomorHpEditText = findViewById(R.id.nomorHpEditText) // Mendapatkan referensi ke EditText untuk nomor telepon dari layout berdasarkan ID alamatEditText = findViewById(R.id.alamatEditText) // Mendapatkan referensi ke EditText untuk alamat dari layout berdasarkan ID registerButton = findViewById(R.id.registerButton) // Mendapatkan referensi ke Button untuk pendaftaran dari layout berdasarkan ID registerButton.setOnClickListener { // Menetapkan OnClickListener untuk tombol registerButton val nama = namaEditText.text.toString() // Mendapatkan teks nama lengkap yang dimasukkan pengguna val email = emailEditText.text.toString() // Mendapatkan teks alamat email yang dimasukkan pengguna val password = passwordEditText.text.toString() // Mendapatkan teks kata sandi yang dimasukkan pengguna val nomorHp = nomorHpEditText.text.toString() // Mendapatkan teks nomor telepon yang dimasukkan pengguna val alamat = alamatEditText.text.toString() // Mendapatkan teks alamat yang dimasukkan pengguna // Validasi sederhana: pastikan semua field tidak kosong </pre>
--	---

	<pre> if (nama.isNotEmpty() && email.isNotEmpty() && password.isNotEmpty() && nomorHp.isNotEmpty() && alamat.isNotEmpty()) { // Memeriksa apakah semua kolom input tidak kosong // Di sini kita akan menyimpan data ke database lokal (Room) nanti // Untuk sekarang, kita simulasikan penyimpanan berhasil Toast.makeText(this, "Registrasi Berhasil", Toast.LENGTH_SHORT).show() // Menampilkan pesan Toast singkat bahwa registrasi berhasil (sementara) // Redirect ke LoginActivity val intent = Intent(this, LoginActivity::class.java) // Membuat Intent untuk berpindah ke aktivitas LoginActivity startActivity(intent) // Memulai aktivitas LoginActivity finish() // Menutup aktivitas RegisterActivity agar pengguna tidak bisa kembali ke halaman registrasi dengan tombol back tanpa logout } else { // Jika ada kolom input yang kosong Toast.makeText(this, "Semua field harus diisi", Toast.LENGTH_SHORT).show() // Menampilkan pesan Toast singkat bahwa semua kolom harus diisi } } } // Fungsi untuk menyimpan data ke database lokal (akan diimplementasikan dengan Room) private fun simpanDataPengguna(nama: String, email: String, password: String, nomorHp: String, alamat: String) { // Fungsi pribadi untuk menyimpan data pengguna (implementasi database akan ditambahkan nanti) // Implementasi penyimpanan data menggunakan Room akan dilakukan nanti // Untuk sekarang, kita hanya menampilkan toast berhasil } } </pre>
4. HomeActivity	<pre> package com.example.lapordesaku // Mendefinisikan nama paket aplikasi import android.content.Intent // Mengimpor kelas Intent untuk berpindah antar aktivitas import android.os.Bundle // Mengimpor kelas Bundle untuk menyimpan dan memulihkan data instance aktivitas import androidx.appcompat.app.AppCompatActivity // Mengimpor kelas AppCompatActivity sebagai kelas dasar </pre>

	<p>untuk aktivitas yang menggunakan fitur tema dan toolbar AppCompatActivity</p> <p>import androidx.cardview.widget.CardView // Mengimpor kelas CardView untuk menampilkan konten dalam panel kartu dengan sudut membulat dan bayangan</p> <p>class HomeActivity : AppCompatActivity() { //</p> <p>Mendefinisikan kelas HomeActivity yang merupakan turunan dari AppCompatActivity</p> <p>private lateinit var tambahPengaduanCard: CardView //</p> <p>Mendeklarasikan variabel untuk CardView yang mengarah ke aktivitas Tambah Pengaduan</p> <p>private lateinit var laporanSayaCard: CardView //</p> <p>Mendeklarasikan variabel untuk CardView yang mengarah ke aktivitas Laporan Saya</p> <p>private lateinit var listPengaduanCard: CardView //</p> <p>Mendeklarasikan variabel untuk CardView yang mengarah ke aktivitas List Pengaduan</p> <p>private lateinit var chatWargaCard: CardView //</p> <p>Mendeklarasikan variabel untuk CardView yang mengarah ke aktivitas Chat Warga</p> <p>private lateinit var tentangAplikasiCard: CardView //</p> <p>Mendeklarasikan variabel untuk CardView yang mengarah ke aktivitas Tentang Aplikasi</p> <p>// private lateinit var logoutCard: CardView // Jika ada menu Logout (dikomentari dalam kode)</p> <p>override fun onCreate(savedInstanceState: Bundle?) { //</p> <p>Metode yang dipanggil saat aktivitas pertama kali dibuat</p> <p>super.onCreate(savedInstanceState) // Memanggil implementasi onCreate dari kelas induk</p> <p>setContentView(R.layout.activity_home) //</p> <p>Menghubungkan layout XML (activity_home.xml) dengan aktivitas ini</p> <p>val toolbar =</p> <p>findViewById<androidx.appcompat.widget.Toolbar>(R.id.toolbar) // Mendapatkan referensi ke Toolbar dari layout berdasarkan ID</p> <p>supportActionBar(toolbar) // Menetapkan Toolbar sebagai action bar untuk aktivitas ini</p> <p>tambahPengaduanCard =</p> <p>findViewById(R.id.tambahPengaduanCard) // Mendapatkan referensi ke CardView Tambah Pengaduan dari layout berdasarkan ID</p> <p>laporanSayaCard =</p> <p>findViewById(R.id.laporanSayaCard) // Mendapatkan</p>
--	---

	<p>referensi ke CardView Laporan Saya dari layout berdasarkan ID</p> <pre> listPengaduanCard = findViewById(R.id.listPengaduanCard) // Mendapatkan referensi ke CardView List Pengaduan dari layout berdasarkan ID chatWargaCard = findViewById(R.id.chatWargaCard) // Mendapatkan referensi ke CardView Chat Warga dari layout berdasarkan ID tentangAplikasiCard = findViewById(R.id.tentangAplikasiCard) // Mendapatkan referensi ke CardView Tentang Aplikasi dari layout berdasarkan ID // logoutCard = findViewById(R.id.logoutCard) // Jika ada menu Logout (mendapatkan referensi ke CardView Logout) tambahPengaduanCard.setOnClickListener { // Menetapkan OnClickListener untuk CardView Tambah Pengaduan startActivity(Intent(this, TambahPengaduanActivity::class.java)) // Membuat dan memulai Intent untuk berpindah ke aktivitas TambahPengaduanActivity } laporanSayaCard.setOnClickListener { // Menetapkan OnClickListener untuk CardView Laporan Saya startActivity(Intent(this, LaporanSayaActivity::class.java)) // Membuat dan memulai Intent untuk berpindah ke aktivitas LaporanSayaActivity } listPengaduanCard.setOnClickListener { // Menetapkan OnClickListener untuk CardView List Pengaduan startActivity(Intent(this, ListPengaduanActivity::class.java)) // Membuat dan memulai Intent untuk berpindah ke aktivitas ListPengaduanActivity } chatWargaCard.setOnClickListener { // Menetapkan OnClickListener untuk CardView Chat Warga startActivity(Intent(this, ChatWargaActivity::class.java)) // Membuat dan memulai Intent untuk berpindah ke aktivitas ChatWargaActivity } </pre>
--	--

	<pre> tentangAplikasiCard.setOnClickListener { // Menetapkan OnClickListener untuk CardView Tentang Aplikasi startActivity(Intent(this, TentangAplikasiActivity::class.java)) // Membuat dan memulai Intent untuk berpindah ke aktivitas TentangAplikasiActivity } // Jika ada menu Logout /*logoutCard.setOnClickListener { // Implementasikan logika logout di sini // Contoh: menghapus sesi login dan kembali ke LoginActivity val sharedPreferences = getSharedPreferences("login_session", MODE_PRIVATE) sharedPreferences.edit().clear().apply() startActivity(Intent(this, LoginActivity::class.java)) finish() }*/ } } </pre>
5. Tambah Pengaduan	<pre> package com.example.lapordesaku // Mendefinisikan nama paket aplikasi import android.content.Intent // Mengimpor kelas Intent untuk berpindah antar aktivitas import android.os.Bundle // Mengimpor kelas Bundle untuk menyimpan dan memulihkan data instance aktivitas import androidx.appcompat.app.AppCompatActivity // Mengimpor kelas AppCompatActivity sebagai kelas dasar untuk aktivitas yang menggunakan fitur tema dan toolbar AppCompatActivity import androidx.cardview.widget.CardView // Mengimpor kelas CardView untuk menampilkan konten dalam panel kartu dengan sudut membulat dan bayangan class HomeActivity : AppCompatActivity() { // Mendefinisikan kelas HomeActivity yang merupakan turunan dari AppCompatActivity private lateinit var tambahPengaduanCard: CardView // Mendeklarasikan variabel untuk CardView yang mengarah ke aktivitas Tambah Pengaduan private lateinit var laporanSayaCard: CardView // Mendeklarasikan variabel untuk CardView yang mengarah ke aktivitas Laporan Saya private lateinit var listPengaduanCard: CardView // Mendeklarasikan variabel untuk CardView yang mengarah ke aktivitas List Pengaduan </pre>

	<pre> private lateinit var chatWargaCard: CardView // Mendeklarasikan variabel untuk CardView yang mengarah ke aktivitas Chat Warga private lateinit var tentangAplikasiCard: CardView // Mendeklarasikan variabel untuk CardView yang mengarah ke aktivitas Tentang Aplikasi // private lateinit var logoutCard: CardView // Jika ada menu Logout (dikomentari dalam kode) override fun onCreate(savedInstanceState: Bundle?) { // Metode yang dipanggil saat aktivitas pertama kali dibuat super.onCreate(savedInstanceState) // Memanggil implementasi onCreate dari kelas induk setContentView(R.layout.activity_home) // Menghubungkan layout XML (activity_home.xml) dengan aktivitas ini val toolbar = findViewById<androidx.appcompat.widget.Toolbar>(R.id.to olbar) // Mendapatkan referensi ke Toolbar dari layout berdasarkan ID supportActionBar(toolbar) // Menetapkan Toolbar sebagai action bar untuk aktivitas ini tambahPengaduanCard = findViewById(R.id.tambahPengaduanCard) // Mendapatkan referensi ke CardView Tambah Pengaduan dari layout berdasarkan ID laporanSayaCard = findViewById(R.id.laporanSayaCard) // Mendapatkan referensi ke CardView Laporan Saya dari layout berdasarkan ID listPengaduanCard = findViewById(R.id.listPengaduanCard) // Mendapatkan referensi ke CardView List Pengaduan dari layout berdasarkan ID chatWargaCard = findViewById(R.id.chatWargaCard) // Mendapatkan referensi ke CardView Chat Warga dari layout berdasarkan ID tentangAplikasiCard = findViewById(R.id.tentangAplikasiCard) // Mendapatkan referensi ke CardView Tentang Aplikasi dari layout berdasarkan ID // logoutCard = findViewById(R.id.logoutCard) // Jika ada menu Logout (mendapatkan referensi ke CardView Logout) tambahPengaduanCard.setOnClickListener { // Menetapkan OnClickListener untuk CardView Tambah Pengaduan </pre>
--	---

	<pre> startActivity(Intent(this, TambahPengaduanActivity::class.java)) // Membuat dan memulai Intent untuk berpindah ke aktivitas TambahPengaduanActivity } laporanSayaCard.setOnClickListener { // Menetapkan OnClickListener untuk CardView Laporan Saya startActivity(Intent(this, LaporanSayaActivity::class.java)) // Membuat dan memulai Intent untuk berpindah ke aktivitas LaporanSayaActivity } listPengaduanCard.setOnClickListener { // Menetapkan OnClickListener untuk CardView List Pengaduan startActivity(Intent(this, ListPengaduanActivity::class.java)) // Membuat dan memulai Intent untuk berpindah ke aktivitas ListPengaduanActivity } chatWargaCard.setOnClickListener { // Menetapkan OnClickListener untuk CardView Chat Warga startActivity(Intent(this, ChatWargaActivity::class.java)) // Membuat dan memulai Intent untuk berpindah ke aktivitas ChatWargaActivity } tentangAplikasiCard.setOnClickListener { // Menetapkan OnClickListener untuk CardView Tentang Aplikasi startActivity(Intent(this, TentangAplikasiActivity::class.java)) // Membuat dan memulai Intent untuk berpindah ke aktivitas TentangAplikasiActivity } // Jika ada menu Logout /*logoutCard.setOnClickListener { // Implementasikan logika logout di sini // Contoh: menghapus sesi login dan kembali ke LoginActivity val sharedPreferences = getSharedPreferences("login_session", MODE_PRIVATE) sharedPreferences.edit().clear().apply() startActivity(Intent(this, LoginActivity::class.java)) finish() }*/ } } </pre>
--	--

<p>6. ListPengaduanActivity</p>	<pre> package com.example.lapordesaku // Mendefinisikan nama paket aplikasi import android.os.Bundle // Mengimpor kelas Bundle untuk menyimpan dan memulihkan data instance aktivitas import android.view.View // Mengimpor kelas View sebagai dasar untuk semua komponen UI import android.widget.TextView // Mengimpor kelas TextView untuk menampilkan teks import androidx.appcompat.app.AppCompatActivity // Mengimpor kelas AppCompatActivity sebagai kelas dasar untuk aktivitas yang menggunakan fitur tema dan toolbar AppCompatActivity import androidx.recyclerview.widget.LinearLayoutManager // Mengimpor kelas LinearLayoutManager untuk mengatur tata letak item dalam RecyclerView secara linear import androidx.recyclerview.widget.RecyclerView // Mengimpor kelas RecyclerView untuk menampilkan kumpulan data dalam tampilan yang efisien import java.text.SimpleDateFormat // Mengimpor kelas SimpleDateFormat untuk memformat tanggal import java.util.Date // Mengimpor kelas Date untuk mendapatkan tanggal saat ini import java.util.Locale // Mengimpor kelas Locale untuk menentukan regionalitas format tanggal class ListPengaduanActivity : AppCompatActivity() { // Mendefinisikan kelas ListPengaduanActivity yang merupakan turunan dari AppCompatActivity private lateinit var semuaLaporanRecyclerView: RecyclerView // Mendeklarasikan variabel untuk RecyclerView yang akan menampilkan semua laporan warga private lateinit var laporanAdapter: LaporanAdapter // Mendeklarasikan variabel untuk adapter yang akan mengelola dan menampilkan data laporan dalam RecyclerView private lateinit var emptyTextView: TextView // Mendeklarasikan variabel untuk TextView yang akan ditampilkan jika tidak ada laporan override fun onCreate(savedInstanceState: Bundle?) { // Metode yang dipanggil saat aktivitas pertama kali dibuat super.onCreate(savedInstanceState) // Memanggil implementasi onCreate dari kelas induk setContentView(R.layout.activity_list_pengaduan) // Menghubungkan layout XML (activity_list_pengaduan.xml) dengan aktivitas ini </pre>
-------------------------------------	---

	<pre> val toolbar = findViewById<androidx.appcompat.widget.Toolbar>(R.id.to olbar) // Mendapatkan referensi ke Toolbar dari layout berdasarkan ID setSupportActionBar(toolbar) // Menetapkan Toolbar sebagai action bar untuk aktivitas ini supportActionBar?.setDisplayHomeAsUpEnabled(true) // Mengaktifkan tombol "kembali" (up navigation) pada action bar semuaLaporanRecyclerView = findViewById(R.id.semuaLaporanRecyclerView) // Mendapatkan referensi ke RecyclerView untuk semua laporan dari layout berdasarkan ID emptyTextView = findViewById(R.id.emptyTextView) // Mendapatkan referensi ke TextView untuk pesan kosong dari layout berdasarkan ID semuaLaporanRecyclerView.layoutManager = LinearLayoutManager(this) // Membuat dan menetapkan LinearLayoutManager untuk mengatur item dalam RecyclerView secara vertikal // Membuat data dummy laporan dari warga lain val dummySemuaLaporan = listOf(// Membuat list berisi objek Laporan dummy dari warga lain Laporan("Drainase Mampet di RT 02", "Infrastruktur", "Air tidak mengalir lancar setelah hujan deras.", getCurrentDate()), // Membuat objek Laporan pertama Laporan("Ada Keributan di Malam Hari", "Keamanan", "Suara gaduh mengganggu ketenangan warga.", getCurrentDate()), // Membuat objek Laporan kedua Laporan("Jadwal Posyandu Bulan Depan", "Layanan Publik", "Informasi mengenai pelaksanaan Posyandu.", getCurrentDate()), // Membuat objek Laporan ketiga Laporan("Lampu Jalan Depan Rumah Mati", "Layanan Publik", "Sudah beberapa hari lampu jalan tidak berfungsi.", getCurrentDate()) // Membuat objek Laporan keempat // Tambahkan data dummy lainnya) if (dummySemuaLaporan.isNotEmpty()) { // Memeriksa apakah list dummySemuaLaporan tidak kosong laporanAdapter = LaporanAdapter(dummySemuaLaporan) // Membuat instance dari LaporanAdapter dengan list dummySemuaLaporan sebagai data </pre>
--	---

	<pre> semuaLaporanRecyclerView.adapter = laporanAdapter // Menetapkan adapter yang telah dibuat ke RecyclerView emptyTextView.visibility = View.GONE // Menyembunyikan TextView untuk pesan kosong semuaLaporanRecyclerView.visibility = View.VISIBLE // Menampilkan RecyclerView yang berisi daftar semua laporan warga } else { // Jika list dummySemuaLaporan kosong semuaLaporanRecyclerView.visibility = View.GONE // Menyembunyikan RecyclerView emptyTextView.visibility = View.VISIBLE // Menampilkan TextView untuk pesan kosong } } private fun getCurrentDate(): String { // Fungsi pribadi untuk mendapatkan tanggal saat ini dalam format dd-MM- yyyy val sdf = SimpleDateFormat("dd-MM-yyyy", Locale.getDefault()) // Membuat instance SimpleDateFormat dengan format tanggal hari-bulan-tahun, menggunakan locale default return sdf.format(Date()) // Memformat objek Date (tanggal saat ini) menjadi string sesuai format dan mengembalikannya } override fun onSupportNavigateUp(): Boolean { // Metode yang dipanggil saat tombol "kembali" pada action bar ditekan onBackPressed() // Memanggil metode onBackPressed() untuk kembali ke aktivitas sebelumnya return true // Mengembalikan true untuk menunjukkan bahwa navigasi "up" telah ditangani } } </pre>
7. Laporan Saya	<pre> package com.example.lapordesaku // Mendefinisikan nama paket aplikasi import android.os.Bundle // Mengimpor kelas Bundle untuk menyimpan dan memulihkan data instance aktivitas import android.view.View // Mengimpor kelas View sebagai dasar untuk semua komponen UI import android.widget.TextView // Mengimpor kelas TextView untuk menampilkan teks import androidx.appcompat.app.AppCompatActivity // Mengimpor kelas AppCompatActivity sebagai kelas dasar untuk aktivitas yang menggunakan fitur tema dan toolbar AppCompatActivity </pre>

	<pre> import androidx.recyclerview.widget.LinearLayoutManager // Mengimpor kelas LinearLayoutManager untuk mengatur tata letak item dalam RecyclerView secara linear import androidx.recyclerview.widget.RecyclerView // Mengimpor kelas RecyclerView untuk menampilkan kumpulan data dalam tampilan yang efisien import java.text.SimpleDateFormat // Mengimpor kelas SimpleDateFormat untuk memformat tanggal import java.util.Date // Mengimpor kelas Date untuk mendapatkan tanggal saat ini import java.util.Locale // Mengimpor kelas Locale untuk menentukan regionalitas format tanggal class LaporanSayaActivity : AppCompatActivity() { // Mendefinisikan kelas LaporanSayaActivity yang merupakan turunan dari AppCompatActivity private lateinit var laporanRecyclerView: RecyclerView // Mendeklarasikan variabel untuk RecyclerView yang akan menampilkan daftar laporan private lateinit var laporanAdapter: LaporanAdapter // Mendeklarasikan variabel untuk adapter yang akan mengelola dan menampilkan data laporan dalam RecyclerView private lateinit var emptyTextView: TextView // Mendeklarasikan variabel untuk TextView yang akan ditampilkan jika tidak ada laporan override fun onCreate(savedInstanceState: Bundle?) { // Metode yang dipanggil saat aktivitas pertama kali dibuat super.onCreate(savedInstanceState) // Memanggil implementasi onCreate dari kelas induk setContentView(R.layout.activity_laporan_saya) // Menghubungkan layout XML (activity_laporan_saya.xml) dengan aktivitas ini val toolbar = findViewById<androidx.appcompat.widget.Toolbar>(R.id.to olbar) // Mendapatkan referensi ke Toolbar dari layout berdasarkan ID setSupportActionBar(toolbar) // Menetapkan Toolbar sebagai action bar untuk aktivitas ini supportActionBar?.setDisplayHomeAsUpEnabled(true) // Mengaktifkan tombol "kembali" (up navigation) pada action bar laporanRecyclerView = findViewById(R.id.laporanRecyclerView) // Mendapatkan referensi ke RecyclerView dari layout berdasarkan ID </pre>
--	---

	<pre> emptyTextView = findViewById(R.id.emptyTextView) // Mendapatkan referensi ke TextView untuk pesan kosong dari layout berdasarkan ID laporanRecyclerView.layoutManager = LinearLayoutManager(this) // Membuat dan menetapkan LinearLayoutManager untuk mengatur item dalam RecyclerView secara vertikal // Membuat data dummy laporan val dummyLaporan = listOf(// Membuat list berisi objek Laporan dummy Laporan("Jalan Rusak di Depan Balai Desa", "Infrastruktur", "Kondisi jalan berlubang sangat membahayakan pengguna jalan.", getDate()), // Membuat objek Laporan pertama Laporan("Sampah Menumpuk di Sungai", "Kebersihan", "Tumpukan sampah menyebabkan bau tidak sedap dan potensi banjir.", getDate()), // Membuat objek Laporan kedua Laporan("Penerangan Jalan Mati", "Layanan Publik", "Lampu jalan mati membuat lingkungan menjadi gelap di malam hari.", getDate()) // Membuat objek Laporan ketiga // Tambahkan data dummy lainnya sesuai kebutuhan) if (dummyLaporan.isNotEmpty()) { // Memeriksa apakah list dummyLaporan tidak kosong laporanAdapter = LaporanAdapter(dummyLaporan) // Membuat instance dari LaporanAdapter dengan list dummyLaporan sebagai data laporanRecyclerView.adapter = laporanAdapter // Menetapkan adapter yang telah dibuat ke RecyclerView emptyTextView.visibility = View.GONE // Menyembunyikan TextView untuk pesan kosong laporanRecyclerView.visibility = View.VISIBLE // Menampilkan RecyclerView yang berisi daftar laporan } else { // Jika list dummyLaporan kosong laporanRecyclerView.visibility = View.GONE // Menyembunyikan RecyclerView emptyTextView.visibility = View.VISIBLE // Menampilkan TextView untuk pesan kosong } } private fun getDate(): String { // Fungsi pribadi untuk mendapatkan tanggal saat ini dalam format dd-MM- yyyy val sdf = SimpleDateFormat("dd-MM-yyyy", Locale.getDefault()) // Membuat instance SimpleDateFormat </pre>
--	---

	<p>dengan format tanggal hari-bulan-tahun, menggunakan locale default</p> <pre> return sdf.format(Date()) // Memformat objek Date (tanggal saat ini) menjadi string sesuai format dan mengembalikannya } override fun onSupportNavigateUp(): Boolean { // Metode yang dipanggil saat tombol "kembali" pada action bar ditekan onBackPressed() // Memanggil metode onBackPressed() untuk kembali ke aktivitas sebelumnya return true // Mengembalikan true untuk menunjukkan bahwa navigasi "up" telah ditangani } } </pre>
8. Laporan	<pre> package com.example.lapordesaku // Mendefinisikan nama paket aplikasi data class Laporan(// Mendefinisikan data class bernama Laporan val judul: String, // Properti untuk menyimpan judul laporan (bertipe String) val kategori: String, // Properti untuk menyimpan kategori laporan (bertipe String) val deskripsi: String, // Properti untuk menyimpan deskripsi laporan (bertipe String) val tanggal: String // Properti untuk menyimpan tanggal laporan (bertipe String)) </pre>
9. Laporan Adapter	<pre> package com.example.lapordesaku // Mendefinisikan nama paket aplikasi import android.view.LayoutInflater // Mengimpor kelas LayoutInflater untuk membuat instance layout XML menjadi objek View import android.view.View // Mengimpor kelas View sebagai dasar untuk semua komponen UI import android.view.ViewGroup // Mengimpor kelas ViewGroup sebagai wadah untuk View dan ViewGroup lainnya import android.widget.TextView // Mengimpor kelas TextView untuk menampilkan teks import androidx.recyclerview.widget.RecyclerView // Mengimpor kelas RecyclerView untuk menampilkan data dalam daftar yang efisien class LaporanAdapter(private val listLaporan: List<Laporan>) : // Mendefinisikan kelas LaporanAdapter yang merupakan turunan dari RecyclerView.Adapter RecyclerView.Adapter<LaporanAdapter.LaporanViewHolder>() { // Menentukan LaporanViewHolder sebagai jenis ViewHolder yang akan digunakan oleh adapter ini </pre>

	<pre> class LaporanViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) { // Mendefinisikan kelas LaporanViewHolder yang merupakan turunan dari RecyclerView.ViewHolder untuk menampung referensi ke View item laporan val judulTextView: TextView = itemView.findViewById(R.id.judulTextView) // Mendapatkan referensi ke TextView untuk judul dari itemView berdasarkan ID val kategoriTextView: TextView = itemView.findViewById(R.id.kategoriTextView) // Mendapatkan referensi ke TextView untuk kategori dari itemView berdasarkan ID val deskripsiTextView: TextView = itemView.findViewById(R.id.deskripsiTextView) // Mendapatkan referensi ke TextView untuk deskripsi dari itemView berdasarkan ID val tanggalTextView: TextView = itemView.findViewById(R.id.tanggalTextView) // Mendapatkan referensi ke TextView untuk tanggal dari itemView berdasarkan ID } override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): LaporanViewHolder { // Metode yang dipanggil saat RecyclerView membutuhkan ViewHolder baru untuk merepresentasikan sebuah item val itemView = LayoutInflater.from(parent.context) // Membuat LayoutInflater dari context parent .inflate(R.layout.item_laporan, parent, false) // Mengembangkan layout XML (item_laporan.xml) menjadi objek View return LaporanViewHolder(itemView) // Membuat dan mengembalikan instance LaporanViewHolder dengan View yang telah dikembangkan } override fun onBindViewHolder(holder: LaporanViewHolder, position: Int) { // Metode yang dipanggil oleh RecyclerView untuk menampilkan data pada posisi tertentu val currentItem = listLaporan[position] // Mendapatkan objek Laporan saat ini dari listLaporan berdasarkan posisi holder.judulTextView.text = currentItem.judul // Mengatur teks pada judulTextView dengan judul dari objek Laporan saat ini holder.kategoriTextView.text = currentItem.kategori // Mengatur teks pada kategoriTextView dengan kategori dari objek Laporan saat ini holder.deskripsiTextView.text = currentItem.deskripsi // Mengatur teks pada deskripsiTextView dengan deskripsi dari objek Laporan saat ini holder.tanggalTextView.text = currentItem.tanggal // Mengatur teks pada tanggalTextView dengan tanggal dari objek Laporan saat ini } override fun getItemCount() = listLaporan.size // Metode yang dipanggil oleh RecyclerView untuk mendapatkan jumlah total item dalam dataset } </pre>
10. Chat Warga	<pre> package com.example.lapordesaku // Mendefinisikan nama paket aplikasi </pre>

	<pre> import android.os.Bundle // Mengimpor kelas Bundle untuk menyimpan dan memulihkan data instance aktivitas import android.widget.Button // Mengimpor kelas Button untuk elemen tombol interaktif import android.widget.EditText // Mengimpor kelas EditText untuk elemen input teks import androidx.appcompat.app.AppCompatActivity // Mengimpor kelas AppCompatActivity sebagai kelas dasar untuk aktivitas yang menggunakan fitur tema dan toolbar AppCompatActivity import androidx.recyclerview.widget.LinearLayoutManager // Mengimpor kelas LinearLayoutManager untuk mengatur tata letak item dalam RecyclerView secara linear import androidx.recyclerview.widget.RecyclerView // Mengimpor kelas RecyclerView untuk menampilkan kumpulan data dalam tampilan yang efisien import java.text.SimpleDateFormat // Mengimpor kelas SimpleDateFormat untuk memformat tanggal dan waktu import java.util.Date // Mengimpor kelas Date untuk mendapatkan tanggal dan waktu saat ini import java.util.Locale // Mengimpor kelas Locale untuk menentukan regionalitas format tanggal dan waktu class ChatWargaActivity : AppCompatActivity() { // Mendefinisikan kelas ChatWargaActivity yang merupakan turunan dari AppCompatActivity private lateinit var chatRecyclerView: RecyclerView // Mendeklarasikan variabel untuk RecyclerView yang akan menampilkan pesan chat private lateinit var pesanAdapter: PesanAdapter // Mendeklarasikan variabel untuk adapter yang akan mengelola dan menampilkan data pesan dalam RecyclerView private lateinit var messageEditText: EditText // Mendeklarasikan variabel untuk EditText tempat pengguna mengetik pesan private lateinit var sendButton: Button // Mendeklarasikan variabel untuk Button yang digunakan untuk mengirim pesan private val listPesan = mutableListOf<Pesan>() // Membuat mutable list untuk menyimpan objek Pesan (data chat) override fun onCreate(savedInstanceState: Bundle?) { // Metode yang dipanggil saat aktivitas pertama kali dibuat super.onCreate(savedInstanceState) // Memanggil implementasi onCreate dari kelas induk setContentView(R.layout.activity_chat_warga) // Menghubungkan layout XML (activity_chat_warga.xml) dengan aktivitas ini val toolbar = findViewById<androidx.appcompat.widget.Toolbar>(R.id.toolbar) // Mendapatkan referensi ke Toolbar dari layout berdasarkan ID setSupportActionBar(toolbar) // Menetapkan Toolbar sebagai action bar untuk aktivitas ini supportActionBar?.setDisplayHomeAsUpEnabled(true) // Mengaktifkan tombol "kembali" (up navigation) pada action bar chatRecyclerView = findViewById(R.id.chatRecyclerView) // Mendapatkan referensi ke RecyclerView dari layout berdasarkan ID messageEditText = findViewById(R.id.messageEditText) // </pre>
--	---

	<p>Mendapatkan referensi ke EditText dari layout berdasarkan ID</p> <pre> sendButton = findViewById(R.id.sendButton) // Mendapatkan referensi ke Button dari layout berdasarkan ID chatRecyclerView.layoutManager = LinearLayoutManager(this) // Membuat dan menetapkan LinearLayoutManager untuk mengatur item dalam RecyclerView secara vertikal pesanAdapter = PesanAdapter(listPesan) // Membuat instance dari PesanAdapter dengan listPesan sebagai data chatRecyclerView.adapter = pesanAdapter // Menetapkan adapter yang telah dibuat ke RecyclerView // Tambahkan beberapa pesan dummy listPesan.add(Pesan("Warga 1", "Halo semua!", getCurrentTime())) // Menambahkan pesan dummy pertama ke listPesan listPesan.add(Pesan("Warga 2", "Hai!", getCurrentTime())) // Menambahkan pesan dummy kedua ke listPesan listPesan.add(Pesan("Warga 1", "Ada info terbaru kah?", getCurrentTime())) // Menambahkan pesan dummy ketiga ke listPesan pesanAdapter.notifyDataSetChanged() // Memberi tahu adapter bahwa data dalam listPesan telah berubah, sehingga RecyclerView akan diperbarui chatRecyclerView.scrollToPosition(listPesan.size - 1) // Menggulir RecyclerView ke posisi item terakhir (pesan terbaru) sendButton.setOnClickListener { // Menetapkan OnClickListener untuk tombol sendButton val message = messageEditText.text.toString().trim() // Mendapatkan teks dari messageEditText, menghapus spasi di awal dan akhir if (message.isNotEmpty()) { // Memeriksa apakah pesan tidak kosong // Di sini kita akan menyimpan pesan ke database lokal (Room) dan mengirimkannya (jika ada fitur online) val newMessage = Pesan("Anda", message, getCurrentTime()) // Membuat objek Pesan baru untuk pesan yang dikirim oleh pengguna listPesan.add(newMessage) // Menambahkan pesan baru ke listPesan pesanAdapter.notifyItemInserted(listPesan.size - 1) // Memberi tahu adapter bahwa ada item baru yang dimasukkan pada posisi terakhir messageEditText.text.clear() // Membersihkan input teks setelah pesan dikirim chatRecyclerView.scrollToPosition(listPesan.size - 1) // Menggulir RecyclerView ke posisi item terakhir (pesan terbaru) } } } private fun getCurrentTime(): String { // Fungsi pribadi untuk mendapatkan waktu saat ini dalam format HH:mm val sdf = SimpleDateFormat("HH:mm", Locale.getDefault()) // Membuat instance SimpleDateFormat dengan format jam dan menit, menggunakan locale default return sdf.format(Date()) // Memformat objek Date (waktu saat </pre>
--	--

	<pre> ini) menjadi string sesuai format dan mengembalikannya } override fun onSupportNavigateUp(): Boolean { // Metode yang dipanggil saat tombol "kembali" pada action bar ditekan onBackPressed() // Memanggil metode onBackPressed() untuk kembali ke aktivitas sebelumnya return true // Mengembalikan true untuk menunjukkan bahwa navigasi "up" telah ditangani } } </pre>
11. Pesan	<pre> package com.example.lapordesaku // Mendefinisikan nama paket aplikasi data class Pesan(// Mendefinisikan data class bernama Pesan val pengirim: String, // Properti untuk menyimpan nama atau ID pengirim pesan (bertipe String) val isi: String, // Properti untuk menyimpan isi atau konten pesan (bertipe String) val waktu: String // Properti untuk menyimpan waktu pengiriman pesan (bertipe String)) </pre>
12. Pesan Adapter	<pre> package com.example.lapordesaku // Mendefinisikan nama paket aplikasi import android.view.LayoutInflater // Mengimpor kelas LayoutInflater untuk membuat instance layout XML menjadi objek View import android.view.View // Mengimpor kelas View sebagai dasar untuk semua komponen UI import android.view.ViewGroup // Mengimpor kelas ViewGroup sebagai wadah untuk View dan ViewGroup lainnya import android.widget.TextView // Mengimpor kelas TextView untuk menampilkan teks import androidx.recyclerview.widget.RecyclerView // Mengimpor kelas RecyclerView untuk menampilkan data dalam daftar yang efisien class PesanAdapter(private val listPesan: List<Pesan>) : // Mendefinisikan kelas PesanAdapter yang merupakan turunan dari RecyclerView.Adapter RecyclerView.Adapter<PesanAdapter.PesanViewHolder>() { // Menentukan PesanViewHolder sebagai jenis ViewHolder yang akan digunakan oleh adapter ini class PesanViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) { // Mendefinisikan kelas PesanViewHolder yang merupakan turunan dari RecyclerView.ViewHolder untuk menampung referensi ke View item pesan val pengirimTextView: TextView = itemView.findViewById(R.id.pengirimTextView) // Mendapatkan referensi ke TextView untuk nama pengirim dari itemView berdasarkan ID val isiTextView: TextView = itemView.findViewById(R.id.isiTextView) // Mendapatkan referensi ke TextView untuk isi pesan dari itemView berdasarkan ID val waktuTextView: TextView = </pre>

	<pre> itemView.findViewById(R.id.waktuTextView) // Mendapatkan referensi ke TextView untuk waktu pesan dari itemView berdasarkan ID } override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PesanViewHolder { // Metode yang dipanggil saat RecyclerView membutuhkan ViewHolder baru untuk merepresentasikan sebuah item val itemView = LayoutInflater.from(parent.context) // Membuat LayoutInflater dari context parent .inflate(R.layout.item_pesan, parent, false) // Mengembangkan layout XML (item_pesan.xml) menjadi objek View return PesanViewHolder(itemView) // Membuat dan mengembalikan instance PesanViewHolder dengan View yang telah dikembangkan } override fun onBindViewHolder(holder: PesanViewHolder, position: Int) { // Metode yang dipanggil oleh RecyclerView untuk menampilkan data pada posisi tertentu val currentItem = listPesan[position] // Mendapatkan objek Pesan saat ini dari listPesan berdasarkan posisi holder.pengirimTextView.text = currentItem.pengirim // Mengatur teks pada pengirimTextView dengan nama pengirim dari objek Pesan saat ini holder.isiTextView.text = currentItem.isi // Mengatur teks pada isiTextView dengan isi pesan dari objek Pesan saat ini holder.waktuTextView.text = currentItem.waktu // Mengatur teks pada waktuTextView dengan waktu pesan dari objek Pesan saat ini } override fun getItemCount() = listPesan.size // Metode yang dipanggil oleh RecyclerView untuk mendapatkan jumlah total item dalam dataset (listPesan) } </pre>
13. Tentang Aplikasi	<pre> package com.example.lapordesaku // Mendefinisikan nama paket aplikasi import android.os.Bundle // Mengimpor kelas Bundle untuk menyimpan dan memulihkan data instance aktivitas import androidx.appcompat.app.AppCompatActivity // Mengimpor kelas AppCompatActivity sebagai kelas dasar untuk aktivitas yang menggunakan fitur tema dan toolbar AppCompatActivity class TentangAplikasiActivity : AppCompatActivity() { // Mendefinisikan kelas TentangAplikasiActivity yang merupakan turunan dari AppCompatActivity override fun onCreate(savedInstanceState: Bundle?) { // Metode yang dipanggil saat aktivitas pertama kali dibuat super.onCreate(savedInstanceState) // Memanggil implementasi onCreate dari kelas induk setContentView(R.layout.activity_tentang_aplikasi) // Menghubungkan layout XML (activity_tentang_aplikasi.xml) dengan aktivitas ini } } </pre>

	<pre>val toolbar = findViewById<androidx.appcompat.widget.Toolbar>(R.id.toolbar) // Mendapatkan referensi ke Toolbar dari layout berdasarkan ID setSupportActionBar(toolbar) // Menetapkan Toolbar sebagai action bar untuk aktivitas ini supportActionBar?.setDisplayHomeAsUpEnabled(true) // Mengaktifkan tombol "kembali" (up navigation) pada action bar }</pre> <pre>override fun onSupportNavigateUp(): Boolean { // Metode yang dipanggil saat tombol "kembali" pada action bar ditekan onBackPressed() // Memanggil metode onBackPressed() untuk kembali ke aktivitas sebelumnya return true // Mengembalikan true untuk menunjukkan bahwa navigasi "up" telah ditangani } }</pre>
--	---