

MODUL API CROSS-PLATFORM

(Proses via. WebView Kotlin)



LAPORAN

Oleh.

IKHSAN AL FARIZI

2023TI015

MUHAMMAD FADILLAH ADITAMA

2023TIP04

GALIH ADJIE RAMADHONI

2023TIK22

BAB I

PENDAHULUAN

1 Latar Belakang

WebView adalah subclass dari class View di Android yang memungkinkan aplikasi menampilkan halaman web secara langsung di dalam antarmuka pengguna aplikasi. Meskipun tidak menyediakan fungsionalitas lengkap seperti browser—misalnya kontrol navigasi atau kolom alamat—WebView memungkinkan Anda merender konten web dalam tata letak sebuah activity.

WebView sangat berguna untuk menampilkan informasi yang sering diperbarui, seperti perjanjian pengguna atau panduan penggunaan. WebView dapat diintegrasikan ke dalam activity dan digunakan untuk memuat halaman web yang dihosting secara online.

Selain itu, WebView merupakan solusi praktis ketika Anda perlu menyajikan konten berbasis internet kepada pengguna, seperti email atau laporan online. Alih-alih melakukan permintaan jaringan, menganalisis data, dan merendernya secara manual menggunakan komponen UI Android, akan lebih efisien jika Anda mendesain halaman web yang responsif dan dioptimalkan untuk perangkat Android, lalu menampilkannya langsung melalui WebView.

2 Tujuan Penulisan

Modul ini bertujuan untuk :

1. Menjelaskan konsep dasar WebView dalam pengembangan aplikasi Android.
2. Memberikan panduan langkah demi langkah dalam mengintegrasikan WebView ke dalam *activity*.
3. Membantu pengembang memahami cara memuat dan menampilkan halaman web melalui WebView.
4. Menunjukkan penerapan WebView dalam kasus penggunaan nyata, seperti menampilkan panduan pengguna atau konten berbasis internet lainnya.
5. Meningkatkan efisiensi pengembangan aplikasi Android dengan memanfaatkan halaman web responsif yang dapat diakses langsung melalui WebView.

3 Pokok permasalahan

- Bagaimana cara menampilkan konten web berbasis API ke dalam aplikasi Android secara efisien tanpa menggunakan pemrosesan JSON langsung?
- Permasalahan integrasi antarmuka web dengan aplikasi untuk alternatif pengembangan cepat.
- Bagaimana solusi praktis menampilkan sistem informasi berbasis web pada aplikasi Android dengan usaha minimum?

BAB II

PEMBAHASAN

1 Pengertian WebView

WebView adalah komponen antarmuka pengguna (UI) yang memungkinkan aplikasi Android menampilkan konten web secara langsung di dalam antarmuka aplikasi tanpa membuka browser eksternal. WebView merupakan turunan dari class View dan berfungsi sebagai jembatan antara aplikasi Android dan situs web. Dengan WebView, pengembang dapat memuat dan menampilkan halaman HTML dari internet (melalui URL) atau yang disimpan secara lokal dalam aplikasi. WebView sangat berguna untuk menyematkan fungsi berbasis web seperti formulir, dokumen HTML, panduan pengguna, dan halaman interaktif tanpa harus membangun semuanya menggunakan elemen native Android.

Secara default, WebView hanya digunakan untuk menampilkan halaman web. Namun, dengan konfigurasi tambahan, pengembang dapat mengaktifkan fitur seperti JavaScript, kontrol navigasi, dan bahkan memungkinkan interaksi antara elemen HTML dan kode Java di Android. Hal ini menjadikan WebView sebagai solusi yang fleksibel untuk mengintegrasikan teknologi web ke dalam aplikasi Android.

2 Fungsi dan Penerapan WebView

WebView memiliki berbagai fungsi dan penerapan dalam pengembangan aplikasi Android, di antaranya:

1. Menampilkan konten web secara langsung di dalam aplikasi tanpa perlu membuka browser eksternal.
2. Integrasi antara aplikasi web dan native misal, aplikasi e-commerce yang menampilkan halaman promosi berbasis HTML.
3. Menampilkan dokumen online sepertihalnya panduan pengguna, kebijakan privasi, dan dokumen legal lainnya.
4. Membuat aplikasi hybrid antara tampilan native dan halaman web yang responsif.
5. Menjalankan halaman HTML lokal sehingga dapat digunakan untuk pengujian, pengembangan, atau dokumentasi secara offline.

3 Kelebihan dan Kekurangan WebView

Kelebihan :

1. Integrasi yang mudah karena dapat digunakan langsung dalam layout XML seperti komponen UI lainnya.
2. Ringan serta fleksibel sehingga cocok untuk konten yang sering berubah dan diperbarui secara berkala.
3. Mendukung HTML, CSS, JavaScript yang memungkinkan pengalaman pengguna yang interaktif dan kaya.
4. Konektivitas konten Dinamis dapat mengambil data dari server web tanpa membuat API terpisah.
5. Cocok untuk aplikasi hybrid yang menggabungkan tampilan WebView dan komponen native Android.

Kekurangan :

1. Performa lebih rendah dibandingkan dengan elemen native Android untuk aplikasi yang kompleks.
2. Jika tidak dikonfigurasi dengan benar, WebView rentan terhadap serangan seperti XSS atau akses URL yang tidak sah.
3. Kurangnya fitur browser lengkap Seperti riwayat, tab, atau toolbar yang harus ditambahkan secara manual jika diperlukan.
4. Ketergantungan pada koneksi internet. Jika konten berasal dari web eksternal, aplikasi mungkin tidak menampilkannya saat offline.
5. Responsivitas dan UI terbatas bergantung pada desain web, bukan sepenuhnya pada sistem Android.

4 Cara Kerja WebView

WebView bekerja dengan memuat dan merender halaman web di dalam tampilan aplikasi. Ketika diberikan URL atau konten HTML, WebView akan menggunakan mesin render WebKit (atau Chromium pada versi Android yang lebih baru) untuk menampilkan halaman tersebut.

Langkah-langkah dasar cara kerja WebView:

1. Aplikasi Android menambahkan komponen `<WebView>` ke dalam layout.
2. WebView diinisialisasi di dalam file Java/Kotlin.
3. URL atau konten HTML diberikan ke WebView menggunakan metode `loadUrl()` atau `loadData()`.
4. WebView akan mengambil dan menampilkan halaman sesuai dengan instruksi yang diberikan.

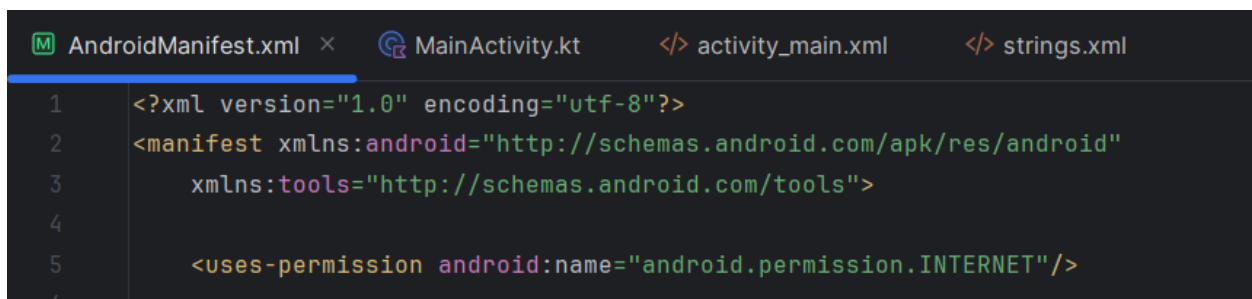
5 Proses pembuatan WebView Menggunakan Android Studio

5.1 Uses Permission

Uses Permission `<uses-permission>` dalam aplikasi diperlukan untuk permintaan izin aplikasi menggunakan fitur ataupun mengakses data sensitive didalam perangkat OS Android.

Beberapa contoh fitur yang harus menggunakan `<uses-permission>` sebagai berikut:

- .INTERNET
- .CAMERA
- .ACCES_FINE_LOCATION
- .READ_EXTERNAL_STORAGE



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools">
4
5     <uses-permission android:name="android.permission.INTERNET"/>
```

Gambar 1

Tanpa `<uses-permission>`, Aplikasi Akan Gagal Mengakses Fitur dikarenakan, WebView butuh INTERNET, tanpa itu system akan memberi pesan error seperti
`net::ERR_CLEARTEXT_NOT_PERMITTED`

5.2 WebView didalam main activity

2.5.1 Baris Import

Apa Kegunaan import di MainActivity.kt? import digunakan untuk mengimpor class, function, atau object dari library lain agar bisa digunakan di file Kotlin. Mirip seperti: **#include** (C++), **import** (Python), **require** (PHP).

Sistem Import yang digunakan ada beberapa untuk dapat memanggil fungsi yang dipakai untuk melakukan aksi menggunakan WebView.

Baris	Fungsi
<code>import android.os.Bundle</code>	Class Bundle untuk menyimpan data saat onCreate()
<code>import android.webkit.WebView</code>	Class WebView, komponen untuk menampilkan halaman web

Kelompok II

<code>import android.webkit.WebViewClient</code>	Agar WebView tidak membuka browser eksternal
<code>import androidx.appcompat.app.AppCompatActivity</code>	Kelas dasar activity modern berbasis Material UI

2.5.2 *Baris MainActivity : AppCompatActivity()*

`AppCompatActivity` adalah subclass dari `FragmentActivity` yang pada gilirannya subclass dari `Activity`. Dengan mewarisi dari `AppCompatActivity`, `MainActivity` mendapatkan banyak fitur penting seperti dukungan `ActionBar`, `fragment`, dan kompatibilitas dengan berbagai versi Android[1].

Dalam pengembangan Android, baris `MainActivity : AppCompatActivity()` mendefinisikan sebuah kelas bernama `MainActivity` yang mewarisi (extends) kelas `AppCompatActivity`. Ini berarti bahwa `MainActivity` mewarisi semua properti dan perilaku `AppCompatActivity` yang disediakan oleh library AndroidX. `AppCompatActivity` biasanya digunakan sebagai kelas dasar untuk sebuah activity karena menyediakan kompatibilitas ke belakang dengan versi Android yang lebih lama dan mendukung berbagai fungsi Android modern seperti `Action Bar`, tema `Material Design`, dan `fragment`.

2.5.3 *Baris @SuppressWarnings()*

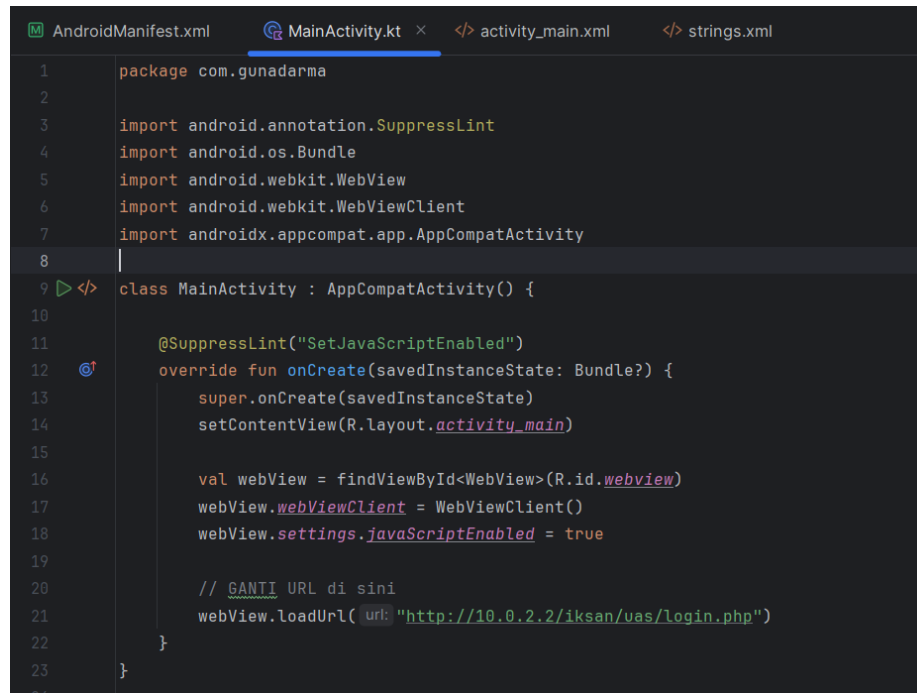
`@SuppressWarnings("SetJavaScriptEnabled")` adalah annotation di Android yang digunakan untuk menonaktifkan peringatan lint yang muncul ketika Anda mengaktifkan JavaScript pada sebuah `WebView` dengan memanggil `setJavaScriptEnabled(true)` pada `WebSettings WebView`[2].

2.5.4 *override fun onCreate()*

`override fun onCreate(savedInstanceState: Bundle?)` adalah metode callback utama dalam siklus hidup (lifecycle) sebuah Activity di Android, yang dipanggil saat Activity pertama kali dibuat oleh system

2.5.5 *webView.loadUrl()*

`webView.loadUrl(String url)` adalah metode yang digunakan untuk memberitahu `WebView` agar memuat dan menampilkan halaman web dari URL yang diberikan[3].



```

1 package com.gunadarma
2
3 import android.annotation.SuppressLint
4 import android.os.Bundle
5 import android.webkit.WebView
6 import android.webkit.WebViewClient
7 import androidx.appcompat.app.AppCompatActivity
8
9 class MainActivity : AppCompatActivity() {
10
11     @SuppressLint("SetJavaScriptEnabled")
12     override fun onCreate(savedInstanceState: Bundle?) {
13         super.onCreate(savedInstanceState)
14         setContentView(R.layout.activity_main)
15
16         val webView = findViewById<WebView>(R.id.webview)
17         webView.webViewClient = WebViewClient()
18         webView.settings.javaScriptEnabled = true
19
20         // GANTI URL di sini
21         webView.loadUrl(uri: "http://10.0.2.2/iksan/uas/login.php")
22     }
23 }

```

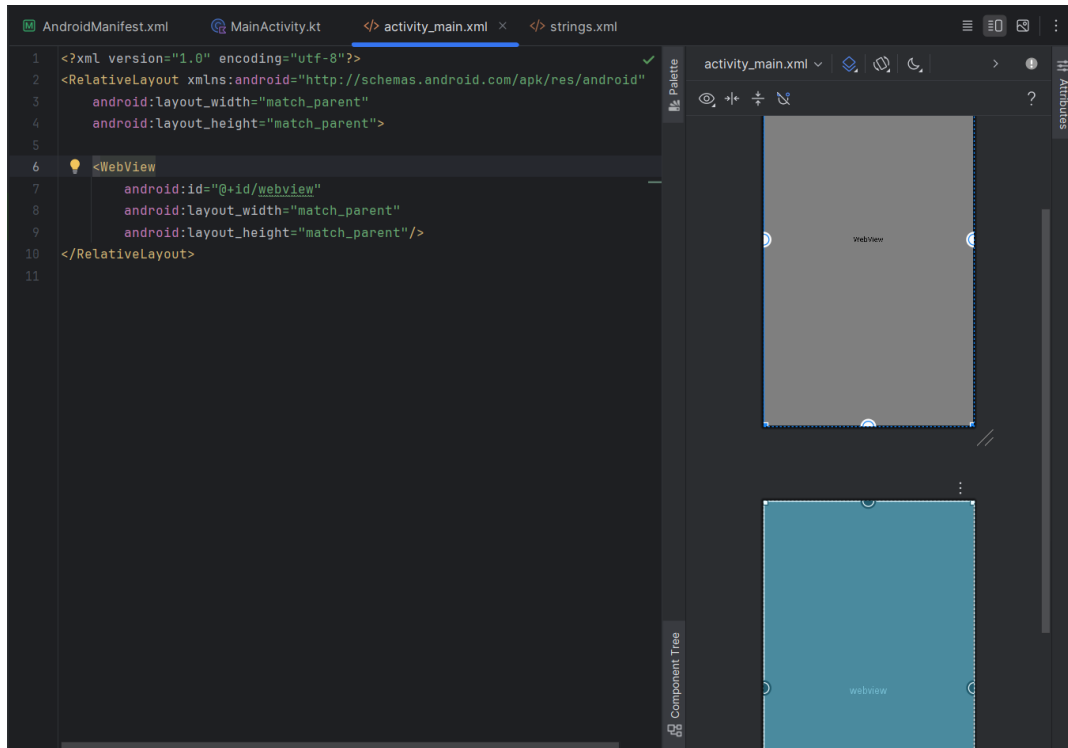
Gambar 2

5.3 WebView didalam activity main

Fungsi WebView di **activity_main.xml** adalah untuk membuat media atau wadah yang menampung wujud dari web itu? Benar sekali, WebView di file **activity_main.xml** berfungsi sebagai wadah (container) untuk menampilkan halaman web di dalam aplikasi Android kamu.

Ini berfungsi untuk container penerima data yang diberikan dari URL yang diambil nantinya dari **MainActivity.kt** makadari itu isi keseluruhan dari **activity_main.xml** berisikan **<WebView>**. Seperti membuat bingkai foto (WebView) lalu meletakkan gambar dari web di dalamnya (**loadUrl()**).

Komponen	Peran
WebView	Tempat menampilkan web dalam aplikasi
loadUrl()	Mengatur URL yang dimuat dalam WebView
webViewClient	Menangani navigasi agar tetap di dalam aplikasi



Gambar 3

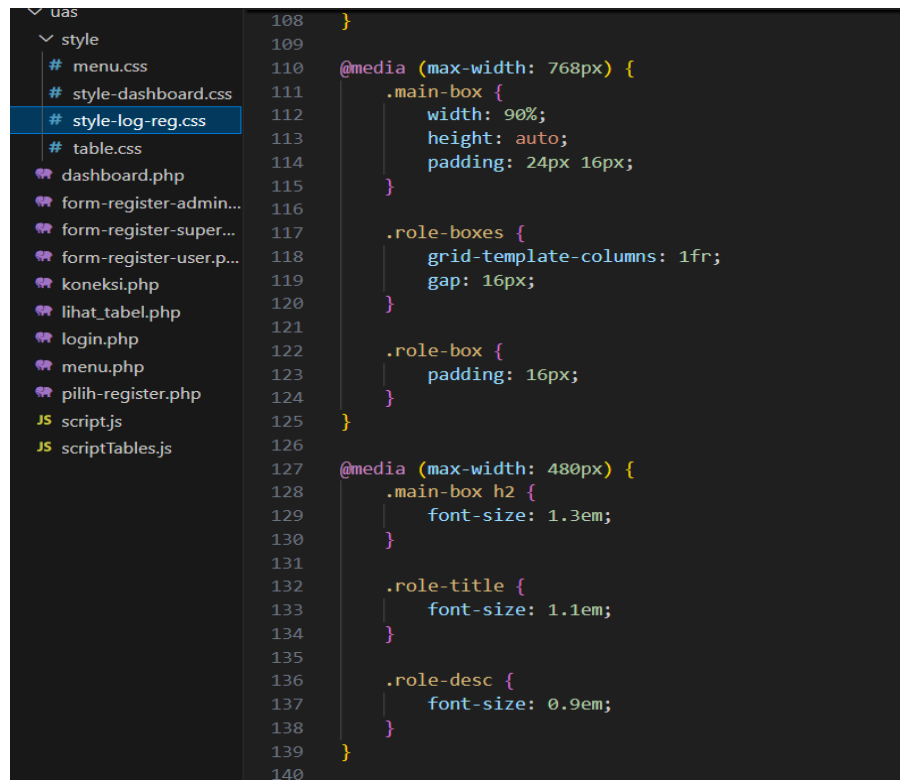
5.4 Pembuatan Style Aplikasi

Media query adalah fitur CSS yang memungkinkan Anda membuat desain responsif dengan mengubah gaya berdasarkan karakteristik perangkat, seperti ukuran layar[4].

max-width: 768px berarti aturan CSS di dalam blok @media tersebut hanya berlaku jika lebar layar kurang dari atau sama dengan 768 piksel[5].

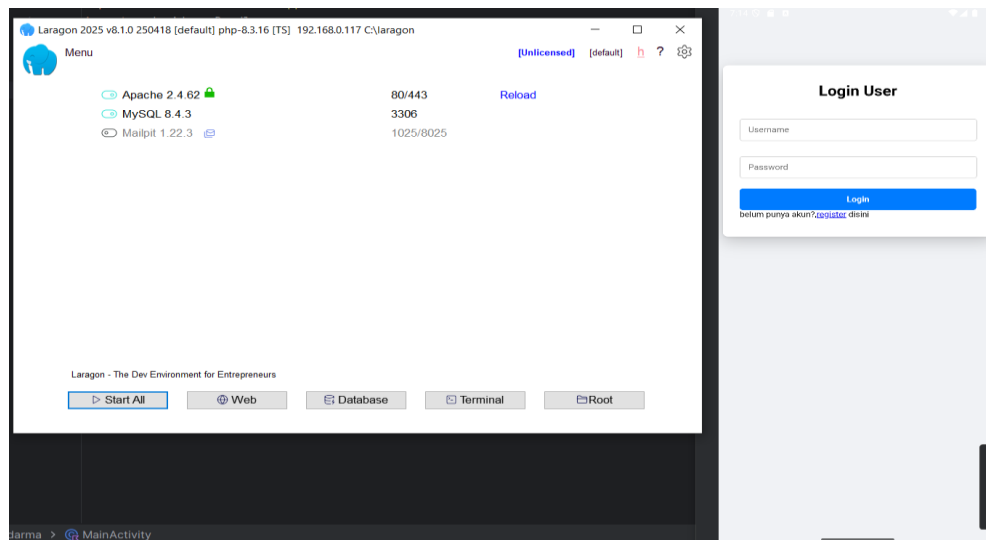
Umumnya, nilai 768px digunakan sebagai batas atas untuk perangkat tablet atau layar kecil, sehingga desain bisa dioptimalkan untuk perangkat tersebut dan yang lebih kecil (misalnya ponsel)[5].

Kelompok II

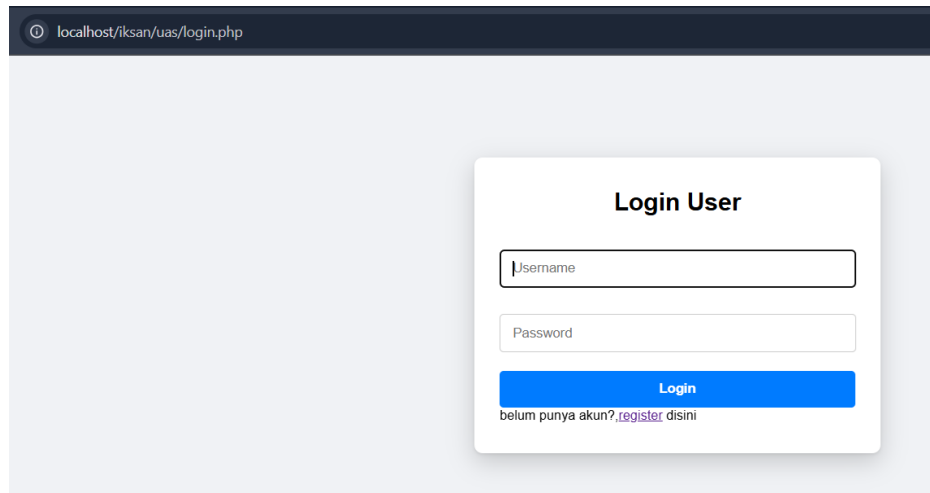


Gambar 4

6 Hasil akhir interface aplikasi



Gambar 5



Gambar 6

BAB III

PENUTUP

1 Kesimpulan

WebView digunakan dalam aplikasi Android untuk memungkinkan developer menampilkan konten web secara langsung di dalam aplikasi tanpa harus membuka browser eksternal. Dalam konteks penggunaan API, WebView dapat digunakan untuk menampilkan antarmuka pengguna (UI) web dan data yang disediakan dari server web, termasuk halaman yang dibangun menggunakan data dari API seperti PHP REST API, tanpa perlu memproses JSON atau XML secara langsung di Android.

Meskipun WebView tidak memberikan kontrol penuh terhadap struktur data seperti saat menggunakan Retrofit atau Volley, metode ini sangat cocok untuk pengembangan yang cepat, terutama ketika backend dan frontend sudah tersedia dalam format berbasis web. Developer cukup memuat halaman web melalui WebView dan mengelola semua perubahan UI dari server web. Hal ini menjadikan WebView sebagai solusi praktis untuk aplikasi informatif, dashboard, atau sistem berbasis web yang perlu diakses melalui Android dengan usaha yang minimal dan waktu pengembangan yang efisien.

2 Saran

Meskipun penggunaan WebView dalam aplikasi Android sangat berguna untuk menampilkan konten web dengan cepat dan efisien terutama untuk sistem internal seperti dashboard kampus atau aplikasi informasi sederhana penggunaannya harus dipertimbangkan dengan hati-hati. WebView direkomendasikan hanya untuk situs web internal atau yang dikelola sendiri, di mana

struktur dan kontennya sepenuhnya dapat dikendalikan dan tidak melibatkan autentikasi yang sensitif. Untuk kebutuhan yang lebih kompleks atau aplikasi yang mencakup data penting seperti transaksi keuangan atau kredensial pengguna, penggunaan WebView sangat tidak disarankan.

Peringatan : WebView rentan terhadap risiko keamanan yang tinggi jika digunakan untuk mengakses situs web publik, terutama jika tidak disertai dengan validasi konten yang tepat, kontrol navigasi, dan pembersihan data. WebView dapat dieksploitasi oleh situs jahat melalui serangan JavaScript injection, phishing, atau Man-in-the-Middle (MITM), terutama saat mengakses konten HTTP yang tidak terenkripsi. Oleh karena itu, penggunaan WebView untuk menampilkan situs web publik dapat menciptakan kerentanan besar terhadap keamanan aplikasi Anda.

MEDIA PELATIHAN

- [1] Jameson, Majid. StackOverflow, Jul 8, 2015 at 15:32. Activity, AppCompatActivity, FragmentActivity, and ActionBarActivity: When to Use Which?, <https://stackoverflow.com/questions/31297246/activity-appcompatactivity-fragmentactivity-and-actionbaractivity-when-to-us>
- [2] Paresh Mayani, November 9, 2013. Issue: Using setJavaScriptEnabled can introduce XSS vulnerabilities, <https://technotalkative.com/issue-using-setjavascriptenabled-can-introduce-xss-vulnerabilities-application-review-carefully/>
- [3] Microsoft Ignite, November 17–21, 2025. WebView.LoadUrl Method, <https://learn.microsoft.com/en-us/dotnet/api/android.webkit.webview.loadurl?view=net-android-35.0>
- [4] w3school. Responsive Web Design - Media Queries, https://www.w3schools.com/css/css_rwd_mediaqueries.asp
- [5] MIMO. CSS Media Query: Syntax, Usage, and Examples, <https://mimo.org/glossary/css/media-query>