

Implementasi *Webview* Menggunakan *Android* Pada *Website* *Cinema Kampus*

M. Ikmal Nurani Mulyasar¹, Alif Amirulloh² dan Arif Mahdi³

STTI Sony Sugema Karawang¹, STTI Sony Sugema Karawang², STTI Sony Sugema Karawang³
Email : ikmalnm17@gmail.com¹, alifamirulloh2001@gmail.com², arifmahdi050902@gmail.com³

ABSTRAKSI

Di tengah perkembangan teknologi yang begitu pesat saat ini, semua orang dapat dengan mudah mengakses berbagai macam hal, salah satunya adalah informasi. Berbagai macam informasi dapat diakses oleh semua orang tak terkecuali informasi terkait satu saluran film atau program TV tertentu. Informasi ini tersebar sangat luas pada *web browser*, hanya dengan melakukan pencarian pada *web browser*, maka informasi terkait film yang dicari akan dapat ditemukan. Namun, hal tersebut masih dianggap merepotkan bagi sebagian orang. Untuk itu, dibuatlah aplikasi *WebView* dari sebuah *website* cinema. *WebView* adalah widget UI yang membantu mengintegrasikan aplikasi *web* ke dalamnya konteks asli aplikasi *Android*. Ini menyediakan mekanisme yang kuat untuk interaksi dua arah antara *native-end* (Java) dan *web-end* (JavaScript) dari aplikasi *Android*. Aplikasi ini bertujuan untuk memudahkan orang untuk mencari informasi terkait film atau konten-konten TV dan meningkatkan pengalaman pengguna dalam menonton film atau konten. Hanya dengan menginstal dan membuka aplikasi ini di *smartphone*, maka seseorang dapat mudah mengetahui film atau konten yang dicari, tanpa repot-repot membuka *link* url *website* film. Dan dari hasil pengujian *blackbox* juga didapatkan hasil 100% yang sesuai dengan apa yang diharapkan.

Kata Kunci: *cinema, film, website, webview*

ABSTRACT

In the midst of today's rapid technological developments, everyone can easily access various things, one of which is information. Everyone can access various kinds of information, including information related to a particular film channel or TV program. This information is spread very widely on web browsers, just by searching on the web browser, information related to the film you are looking for can be found. However, this is still considered a hassle for some people. For this reason, a *WebView* application was created from a cinema website. *WebView* is a UI widget that helps integrate web applications into the native context of *Android* applications. It provides a powerful mechanism for two-way interaction between the *native-end* (Java) and *web-end* (JavaScript) of *Android* applications. This application aims to make it easier for people to find information related to films or TV content and improve the user's experience in watching films or content. Just by installing and opening this application on a smartphone, someone can easily find out the film or content they are looking for, without having to bother opening the film website URL link. And from the black box testing results, 100% results were obtained which were in line with what was expected.

Keywords: *cinema, film, website, webview*

Penulis Korespondensi

M. Ikmal Nurani M

Tanggal Submit : 13/06/2024

Tanggal Diterima : 14/12/2024

Tanggal Terbit : 28/02/2025

This is an open access article under the [CC-BY-NC-SA](https://creativecommons.org/licenses/by-nc-sa/4.0/) license



Copyright: © 2023 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 (CC BY-NC-SA 4.0) International License (<https://creativecommons.org/licenses/by-nc-sa/4.0/>).

Publisher's Note: JPPM stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

I. PENDAHULUAN

Aplikasi berbasis Mobie yang berkembang pesat. Perkembangan ini memungkinkan terjadinya perubahan kebiasaan atau penambahan kebiasaan baru, dan masyarakat kini semakin banyak yang menggunakan gadget dalam melakukan aktivitasnya. Penggunaan

gadget sangatlah penting dalam mendukung aktivitas [1].

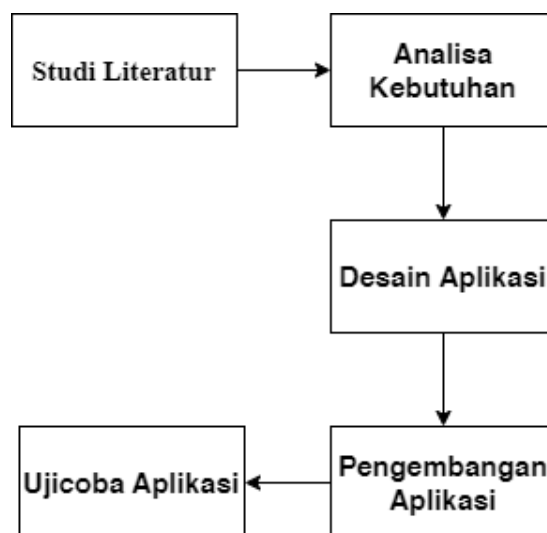
WebView adalah komponen penting di *Android* dan *iOS* platform, memungkinkan aplikasi ponsel cerdas dan tablet untuk menyematkan a *browser* sederhana namun kuat di dalamnya. Untuk mencapai interaksi yang lebih baik antara aplikasi dan “*browser*” yang

tertanam di dalamnya. *WebView* menyediakan sejumlah API (*Application Programming Interface*), mengaktifkan kode di aplikasi Anda untuk memanggil dan dipanggil dengan kode JavaScript di halaman *web*, mencegat kejadiannya, dan memodifikasinya acara. Dengan menggunakan fitur ini, aplikasi dapat disesuaikan "*browser*" untuk aplikasi *web* yang dimaksudkan. Saat ini, di pasar Android, 86 persen dari 20 aplikasi yang paling banyak diunduh dalam 10 kategori berbeda menggunakan *WebView*. [2]

Ada dua jenis API di *WebView*, yang berbasis *Web* API dan API berbasis UI (*User Interface*). API berbasis *web* adalah dirancang untuk berinteraksi dengan konten *web* di dalamnya tampilan *Web*. Contoh API ini termasuk *loadURL*, *CookieManager*, *getCookie*, dan lain-lain. Serangan yang dijelaskan dalam menargetkan API berbasis *Web*. *WebView* adalah subkelas dari kelas *View* yang lebih umum, sedangkan *View* adalah kelas dasar untuk *widget* yang biasa digunakan membuat komponen UI interaktif (tombol, kolom teks, dan lain-lain). Oleh karena itu, *WebView* mewarisi API kelas super. API tersebut adalah API berbasis UI. Serangan dijelaskan dalam target API berbasis UI [3].

Untuk mempermudah pembuatan aplikasi Android, ada teknik yang disebut templat. Itu digunakan untuk memfasilitasi pemahaman konseptual tentang konsep pemrograman berdasarkan template. Bagaimana templat video game mendukung dan memandu pemahaman konseptual pemrograman berorientasi objek (PBO) dan bagaimana user menggunakannya untuk membuat *game* multimedia interaktif mereka sendiri. Dengan menggunakan templat ini, penguji menemukan tiga tingkat interaksi yang menghasilkan pemahaman konseptual yang lebih baik. Pertama, umpan balik visual dari permainan, khususnya perilaku tak terduga, membawa perhatian pada kekurangan dalam desain perangkat lunak. Ketika siswa mengalami kesalahan ini, mereka mulai tingkat lain interaksi dengan *template*. Kedua adalah untuk Jelajahi sumber daya templat. Eksplorasi ini dapat mengarah pada penemuan informasi penting dan hubungan antara sumber daya dan kode model yang nantinya dapat digunakan. Ketiga, ketika siswa bereksplorasi, mereka mulai meningkatkan pemahaman mereka tentang kode dan kode. Pada tingkat kognitif, template meningkatkan pemahaman konseptual karena memberikan siswa landasan atau basis kode yang tidak lengkap untuk membangun program. Berdasarkan uraian tersebut, dapat kita simpulkan bahwa templat dapat mempermudah proses pembelajaran dan pengembangan program [4].

II. METODE PENELITIAN



Gambar 1 Metodologi Penelitian

- 1) Studi literatur: Studi literatur adalah suatu metode pengumpulan data-data atau sumber-sumber yang berkaitan dengan topik yang sedang dibahas dalam penelitian [5]. Kami akan melakukan tinjauan literatur pada *website* bioskop kampus, termasuk pengoperasiannya, kelebihan dan kekurangannya, serta cara menggunakannya dalam pengembangan aplikasi. Penelitian literatur dapat dilakukan dengan membaca buku, jurnal akademik, artikel teknis, dan sumber *online* lainnya.
- 2) Analisis kebutuhan pengguna: Melakukan analisis kebutuhan pengguna untuk mengetahui fitur-fitur apa saja yang dibutuhkan dalam aplikasi berbasis *WebView*. Pengembang dapat melakukan wawancara dengan pengguna potensial atau melakukan survei untuk mengetahui kebutuhan pengguna secara langsung. Analisis kebutuhan pengguna menjadi sangat penting karena merupakan dasar desain. Ini dirancang untuk menghindari proses implementasi yang buruk dari sistem *elearning*, yang terutama karena tidak relevannya sistem yang dikembangkan dengan kebutuhan nyata pengguna [6].
- 3) Uji coba aplikasi: adalah serangkaian kegiatan untuk menerapkan atau mengimplementasikan sistem melalui sejumlah aktifitas. Dari uji coba ini diharapkan kelemahan dan kelebihan sistem dapat diketahui [7]. Melakukan uji coba aplikasi berbasis *WebView* dengan menggunakan perangkat Android yang berbeda-beda. Pengembang harus memastikan bahwa aplikasi berbasis *WebView* berfungsi dengan baik di berbagai perangkat dan dalam berbagai kondisi jaringan serta uji coba atau pengujian ini dilakukan dengan 2 cara yaitu pengujian *Whitebox* dan pengujian *Blackbox*. *Whitebox* testing adalah suatu metode pengujian aplikasi yang menggunakan penjelasan struktur kontrol sebagai bagian dari component-level design untuk membuat test cases. *White Box* sendiri mempunyai beberapa teknik di dalam pengujiannya, seperti: *Data Flow Testing*, *Control Flow Testing*, *Basic Path / Path Testing*, dan *Loop Testing* [8]. Pengujian *Blackbox* merupakan sebuah pengujian yang digunakan untuk melengkapi pengujian sebelumnya yaitu *Whitebox* Testing agar aplikasi yang kita buat

memiliki kualitas yang baik serta waktu yang digunakan akan lebih efektif, sehingga dapat menguntungkan bagi perusahaan [9].

3. Tampilan *Webview*

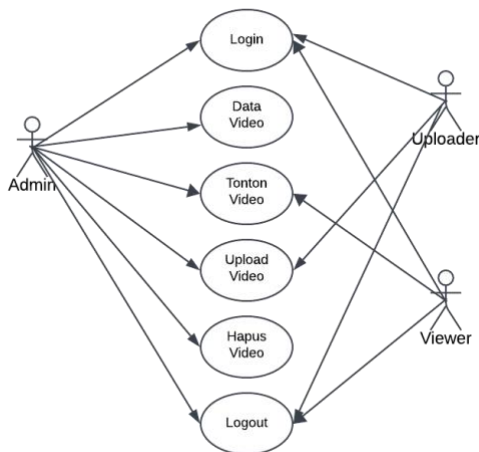
III. HASIL DAN PEMBAHASAN

Pada pengimplementasian *WebView* pada *Website Cinema Kampus* ini, terdapat empat aktor yang terlibat yaitu admin, *uploader*, *viewer* dan superadmin. Tabel 2 menjelaskan fungsi-fungsi dari masing-masing aktor.

Tabel 1 Tabel Aktor Yang Terlibat

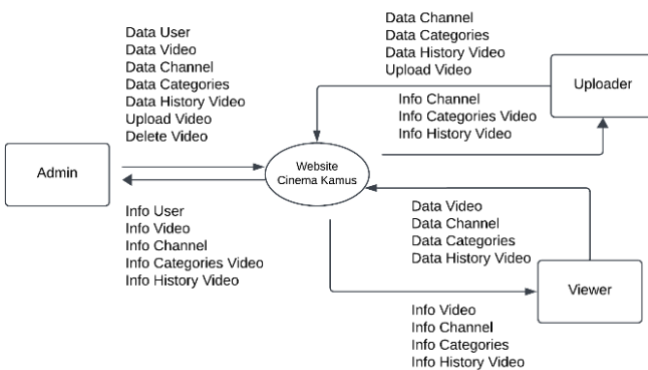
No	Aktor yang terlibat	Keterangan
1	Admin	Dapat mengakses semua data-data yang terdapat disistem termasuk dapat mengedit dan menghapus video-video.
2	Uploader	Dapat mengupload video ke dalam sistem
3	Viewer	Dapat melihat konten yang ada didalam sistem dan dapat mengikuti akun pemilik video.

1. Usecase

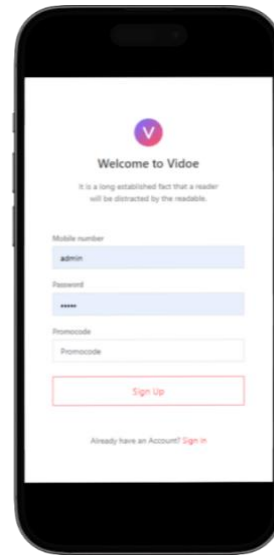


Gambar 2 Usecase Website Cinema Kampus

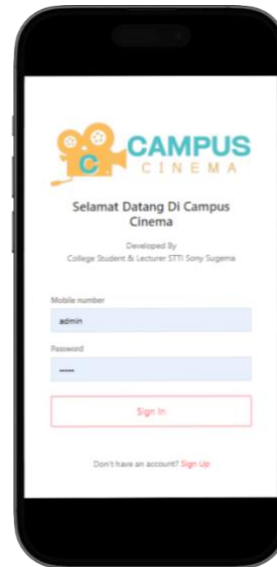
2. Konteks Digram



Gambar 3 Konteks Diagram



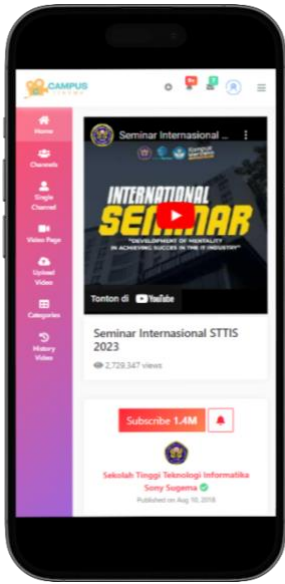
Gambar 4 Tampilan Daftar



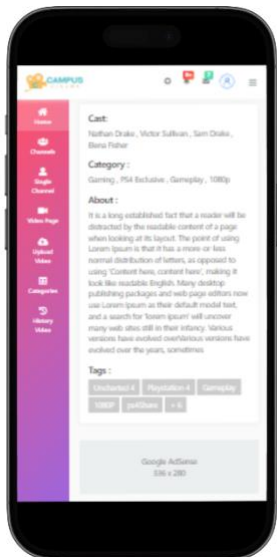
Gambar 5 Tampilan Login

Gambar 4 memperlihatkan tampilan register untuk para pengguna untuk mendaftarkan akun. Ketika sudah mendaftarkan akun, akan langsung diarahkan untuk login ke dalam sistem dan pada gambar 5 memperlihatkan tampilan login ketika

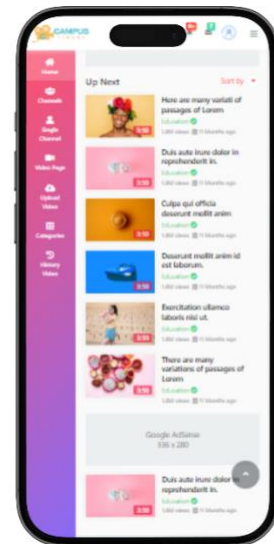
sudah mendaftar.



Gambar 6 Tampilan Dashboard

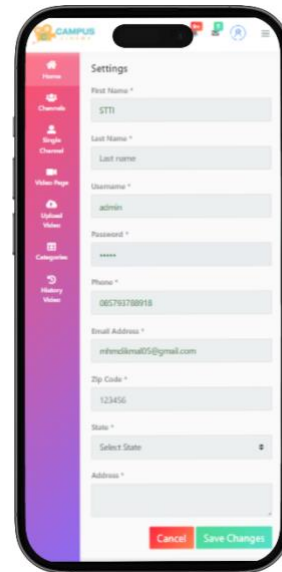


Gambar 7 Tampilan Dashboard



Gambar 8 Tampilan Dashboard

Pada Gambar 6 sampai gambar 8 adalah tampilan *dashboard* atau tampilan utama pada halaman admin dan terdapat beberapa menu diantaranya *Channels*, *Single Channel*, *Video Page*, *Upload Video*, *Categories*, dan *History Video*



Gambar 9 Tampilan Update Data dan Setting User

Pada Gambar 9 ini memperlihatkan tampilan *setting* untuk mengupdate atau mengubah data-data akun seperti nama, *email* dan lain sebagainya.

4. Pengujian Sistem

Pada Pengujian ini menggunakan pengujian Blackbox.

Tabel 2 Pengujian Blackbox

No	Skenario Pengujian	Uji Kasus
1	Halaman Daftar	Semua Data Terisi
2	Halaman Login	Username (diisi) Password (diisi)
3	Halaman Channel	Masuk ke halaman Channel

4	Halaman Single Video	Masuk ke halaman Single Video
5	Halaman Video Page	Masuk ke halaman Video Page
6	Halaman History Video	Masuk ke halaman History Video
7	Halaman Update Data di Setting User	Mengubah Data User

Hasil Yang Diharapkan	Hasil Pengujian
Berhasil mendaftar	Sesuai Harapan
Berhasil Masuk Website	Sesuai Harapan
Berhasil masuk halaman channels	Sesuai Harapan
Berhasil masuk	Sesuai Harapan
Berhasil Masuk	Sesuai Harapan
Berhasil Masuk	Sesuai Harapan
Berhasil merubah	Sesuai Harapan

IV. KESIMPULAN

Berdasarkan dari hasil pembahasan dapat diambil kesimpulan dari penelitian dan pembuatan *WebView Website Cinema Kampus* yaitu, Analisis yang dilakukan menunjukkan pengimplementasian *WebView* pada sistem *Website Cinema Kampus* dan perlu dikembangkan khususnya agar sistem bisa diakses dan dilihat dari sisi *Android*. Aplikasi yang dibuat tetap mempertahankan fungsi-fungsi yang sudah berjalan didalam sistem *Website Cinema Kampus*. Dan dari hasil pengujian *blackbox* juga didapatkan hasil 100% yang sesuai dengan apa yang diharapkan.

DAFTAR PUSTAKA

- [1] A. Hidayat and N. A. Amri, "Implementasi aplikasi profil institusi berbasis android program studi bisnis internasional dan pemasaran digital," *Jurnal Manajemen Informatika*, vol. 9, no. 1, pp. 11–20, 2022.
- [2] T. Luo, H. Hao, W. Du, Y. Wang, and H. Yin, "Attacks on WebView in the Android system," *ACM International Conference Proceeding Series*, pp. 343–352, 2011, doi: 10.1145/2076732.2076781.
- [3] A. Bhavani, "Cross- Cross - site Scripting Attacks on Android WebView," *IJCSN International Journal of Computer Science and Network*, vol. 2, no. 2, pp. 1–5, 2013.
- [4] R. B. D. Putra, E. S. Budi, and A. R. Kadafi, "Perancangan WebView Template Pada Android Studio Arctic Fox Studi Kasus: Dagangrumah.com," *JURIKOM (Jurnal Riset Komputer)*, vol. 8, no. 6, pp. 374–381, Dec. 2021, doi: 10.30865/jurikom.v8i6.3685.
- [5] B. A. Habsy, "Seni Memahami Penelitian Kuliitatif Dalam Bimbingan Dan Konseling : Studi Literatur," *Jurnal Konseling Andi Matappa*, vol. 1, no. December, pp. 90–100, 2020, doi: 10.235678/25271987.
- [6] R. A. Azmi and H. Maksum, "ANALISIS KEBUTUHAN PENGEMBANGAN MEDIA PEMBELAJARAN BERBASIS WEB MATA PELAJARAN ADMINISTRASI INFRASTRUKTUR JARINGAN," *Jurnal Imiah Pendidikan dan Pembelajaran*, vol. 4, pp. 303–314, 2020.
- [7] A. Prajana, "Pemanfaatan Aplikasi Whatsapp Untuk Media Pembelajaran Dalam Lingkungan Uin Ar-Raniry Banda Aceh," *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, vol. 1, no. 2, p. 122, 2017, doi: 10.22373/cs.v1i2.1980.
- [8] M. Farhan Londjo, "IMPLEMENTASI WHITE BOX TESTING DENGAN TEKNIK BASIS PATH PADA

- PENGUJIAN FORM LOGIN," *Jurnal Siliwangi Seri Sains dan Teknologi*, vol. 7, no. 2, pp. 35–40, 2021.
- [9] F. C. Ningrum, D. Suherman, S. Aryanti, H. A. Prasetya, and A. Saifudin, "Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions," vol. 4, no. 4, pp. 125–130, 2019, [Online]. Available: <http://openjournal.unpam.ac.id/index.php/informatika>