

# UNIVERSITAS DIPONEGORO

**PEMBUATAN APLIKASI “JMN ANYWHERE” SEBAGAI**

**APLIKASI *LIVE VIDEO STREAMING* PADA**

**PT SIMS JOGJA MEDIANET BERBASIS ANDROID**

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat**

**untuk memperoleh gelar Sarjana Teknik**

**AKHMAD JUNAEDI**

**21120112140085**

**FAKULTAS TEKNIK**

**DEPARTEMEN TEKNIK SISTEM KOMPUTER**

**SEMARANG**

**JUNI 2017**

# HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh

Nama : Akhmad Junaedi

NIM : 21120112140085

Jurusan/Program Studi : Sistem Komputer

Judul Tugas Akhir : Pembuatan Aplikasi “JMN Anywhere” Sebagai Aplikasi *Live Video Streaming* pada PT SIMS Jogja Medianet Berbasis Android

**Telah berhasil dipertahankan di hadapan Tim Penguji dan diterima sebagai persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Jurusan / Program Studi Sistem Komputer, Fakultas Teknik, Universitas Diponegoro.**

**TIM PENGUJI**

Pembimbing I : Agung Budi Prasetijo, S.T., M.IT., Ph.D. ( )

Pembimbing II : Ike Pertiwi Windasari, S.T., M.T. ( )

Ketua Penguji : Ir. Kodrat Iman Satoto, M.T. ( )

Sekretaris Penguji : Dania Eridani, S.T., M.Eng. ( )

Anggota Penguji : Adnan Fauzi, S.T., M.Kom. ( )

|  |
| --- |
| Semarang, 14 Juni 2017  Ketua Departemen Teknik Sistem Komputer  Dr. R. Rizal Isnanto, S.T., M.M., M.T.  NIP. 197007272000121001 |

# HALAMAN PERNYATAAN ORISINALITAS

**Tugas Akhir ini adalah karya saya sendiri, dan semua sumber baik yang dikutip maupun yang dirujuk telah saya nyatakan dengan benar.**

NAMA : AKHMAD JUNAEDI

NIM : 21120112140085

Tanda Tangan :

Tanggal : Semarang, 06 Juni 2017

# HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Universitas Diponegoro, saya yang bertandatangan di bawah ini :

Nama : AKHMAD JUNAEDI

NIM : 21120112140085

Jurusan/Program Studi : SISTEM KOMPUTER

Fakultas : TEKNIK

Jenis Karya : TUGAS AKHIR

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Diponegoro **Hak Bebas Royalti Noneksklusif** *(None-exclusive Royalty Free Right)* atas karya ilmiah saya yang berjudul:

**Pembuatan Aplikasi “JMN Anywhere” Sebagai Aplikasi *Live Video Streaming* pada PT SIMS Jogja Medianet Berbasis Android**

beserta perangkat yang ada (jika diperlukan). Dengan ini Hak Bebas Royalti/Noneksklusif ini. Universitas Diponegoro berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Semarang

Pada Tanggal : 16 Juni 2017

Yang menyatakan

Akhmad Junaedi

# ABSTRAK

*Di era moderen, banyak bermunculan inovasi-inovasi baru dalam dunia teknologi. Seperti halnya masyarakat dalam mendapatkan informasi-informasi yang terkini dari berbagai penjuru dunia. Mereka dengan mudah mengakses Internet melalui telepon pintar mereka. Hal itu membuat masyarakat mempunyai ketergantungan yang tinggi terhadap telepon pintar. Oleh karenanya, masyarakat jarang menonton siaran televisi secara langsung melalui televisi untuk mendapatkan informasi terkini. Sehingga, dibutuhkan suatu aplikasi* live video streaming *yang digunakan khusus untuk media penyiaran televisi. Aplikasi ini dirancang untuk dapat menayangkan* live video streaming *berbasis Android yang dapat digunakan sebagai media penyiaran televisi melalui koneksi Internet. Sehingga, masyarakat khususnya para pelanggan PT SIMS Jogja Medianet dapat mengakses siaran televisi secara langsung dimana saja dan kapan saja melalui telepon pintar yang mereka miliki.*

*Aplikasi* live video streaming *ini dibuat menggunakan Android Studio IDE, Java, PHP, dan basis data* *MySQL. Metode pengembangan sistem yang digunakan adalah* waterfall *yang terdiri dari analisa sistem, desain sistem, implementasi dan pengujian. Pengujian fungsional dilakukan dengan menggunakan metode pengujian* Black-Box.

*Hasil dari penelitian ini adalah sebuah aplikasi berbasis Android yang digunakan sebagai media penyiaran televisi. Aplikasi ini dapat digunakan oleh masyarakat khususnya para pelanggan PT SIMS Jogja Medianet untuk mengakses siaran televisi secara langsung melalui telepon pintar dengan koneksi Internet. Semua fungsi pada aplikasi ini sudah dilakukan pengujian dan berjalan sesuai dengan fungsionalitasnya masing-masing. Aplikasi* live video streaming *ini dijalankan pada telepon pintar berbasis Android dengan sistem operasi versi 4.2 (*Jelly Bean*) hingga versi 7.1 (*Nougat)*.*

***Kata Kunci: Video Streaming; Android; Java; PHP***

# ABSTRACT

*In the modern era, innovation in technology has been mushroomed. For example, people change the way in getting the latest information from around the world. They easily access the Internet through their smartphones. Therefore, this creates people have a high dependency on smartphones. Therefore, people rarely watch live broadcast via televisions to get such information. Thus, there is a need of live video streaming application that is used specifically for television broadcast media.This application has been designed as a live video streaming application based on Android, which can be used as a television broadcasting media through the Internet. Thus, the people especially the customers of PT SIMS Jogja Medianet can access live television broadcast anywhere and anytime via smartphones.*

*This live video streaming application is built with the use of Android Studio IDE, Java, PHP, and MySQL database. System development method used is waterfall which consists of system analysis, system design, implementation and testing blocks. Functional testing is done by using Black-Box testing method.*

*The results of this research is an Android based application that can be used as a media for broadcasting television. The application can be used by the people, especially customers of PT SIMS Jogja Medianet to access live television broadcast via smartphones through the Internet. All functions in this application have been tested and running. This live streaming video application run on Android based smartphones with operating system version 4.2 (Jelly Bean) up to version 7.1 (Nougat).*

***Keywords*** *:* ***Video Streaming; Android; Java; PHP***

# KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul “**Pembuatan Aplikasi “JMN Anywhere” Sebagai Aplikasi *Live Video Streaming* pada PT SIMS Jogja Medianet Berbasis Android**”.

Dalam proses pengerjaan Tugas Akhir ini penulis banyak mendapatkan dukungan, bimbingan, bantuan, serta doa dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada:

1. Dr. R. Rizal Isnanto, S.T., M.M., M.T. selaku Ketua Departemen Teknik Sistem Komputer Fakultas Teknik Universitas Diponegoro.
2. Ike Pertiwi Windasari, S.T., M.T. selaku Koordinator Tugas Akhir, sekaligus sebagai dosen wali Penulis yang telah membimbing penulis selama Penulis menjalani masa studi di Departemen Teknik Sistem Komputer Universitas Diponegoro.
3. Agung Budi Prasetijo, S.T., M.IT., Ph.D dan Ike Pertiwi Windasari, S.T., M.T. selaku dosen pembimbing yang telah memberikan masukan, petunjuk, dan bimbingan selama pembuatan Tugas Akhir ini.
4. Bapak dan Ibu dosen Departemen Teknik Sistem Komputer Universitas Diponegoro atas ilmunya yang selama ini telah diberikan.
5. Kedua orang tua dan kedua kakak yang Penulis cintai yang memberikan waktunya untuk membimbing, mendoakan dan memberikan motivasi bagi penulis selama penyelesaian Tugas Akhir ini.
6. Sahabat terbaik Penulis sampai sekarang yaitu Wahid Marlisa, Udin, Dimas, Fian, Ati, dan Ikhsan yang selalu mendukung dan mendoakan sampai Penulis menyelesaikan Tugas Akhir ini.
7. Teman seperjuangan terbaik Penulis yaitu Danang, Fajar, Bondan, Riski, dan Misbah yang selalu mendukung, mendoakan, dan menemani sampai Penulis menyelesaikan Tugas Akhir ini.
8. Seluruh teman-teman Teknik Sistem Komputer, khususnya teman-teman seperjuangan angkatan 2012 yang saling mendukung dan mendoakan proses penyelesaian Tugas Akhir masing-masing.

Dalam penyusunan Tugas Akhir ini, Penulis menyadari bahwa masih terdapat kekurangan karena keterbatasan waktu dan ilmu pengetahuan. Oleh karena itu, Penulis menerima kritik dan saran yang bersifat membangun dari pembaca guna penyusunan Tugas Akhir yang lebih baik lagi di masa yang akan datang. Penulis berharap semoga Tugas Akhir ini dapat bermanfaat baik bagi diri Penulis sendiri maupun bagi masyarakat luas.

Semarang, 04 Juni 2017

Penulis

# DAFTAR ISI

[UNIVERSITAS DIPONEGORO i](#_Toc484596725)

[HALAMAN PENGESAHAN ii](#_Toc484596726)

[HALAMAN PERNYATAAN ORISINALITAS iii](#_Toc484596727)

[HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI iv](#_Toc484596728)

[ABSTRAK v](#_Toc484596729)

[ABSTRACT vi](#_Toc484596730)

[KATA PENGANTAR vii](#_Toc484596731)

[DAFTAR ISI ix](#_Toc484596732)

[DAFTAR GAMBAR xi](#_Toc484596733)

[DAFTAR TABEL xiii](#_Toc484596734)

[BAB I PENDAHULUAN 1](#_Toc484596735)

[1.1 Latar Belakang 1](#_Toc484596736)

[1.2 Rumusan Masalah 2](#_Toc484596737)

[1.3 Batasan Masalah 2](#_Toc484596738)

[1.4 Tujuan Penelitian 3](#_Toc484596739)

[1.5 Manfaat Penelitian 3](#_Toc484596740)

[1.5.1 Manfaat bagi Penulis 3](#_Toc484596741)

[1.5.2 Manfaat bagi pengguna perangkat lunak 3](#_Toc484596742)

[1.6 Sistematika Penulisan 4](#_Toc484596743)

[BAB II LANDASAN TEORI 5](#_Toc484596744)

[2.1 Penelitian Terdahulu 5](#_Toc484596745)

[2.2 *Video Streaming* 6](#_Toc484596746)

[2.3 Gambaran Kerja *Video Streaming* 6](#_Toc484596747)

[2.4 Teknologi *Video Streaming* 9](#_Toc484596748)

[2.4.1 Internet Protokol 9](#_Toc484596749)

[2.4.2 *Server Stream Distribution* 10](#_Toc484596750)

[BAB III PERANCANGAN SISTEM 12](#_Toc484596751)

[3.1 Analisis Kebutuhan 13](#_Toc484596755)

[3.1.1 Kebutuhan Fungsional 13](#_Toc484596756)

[3.1.2 Kebutuhan Non-Fungsional 14](#_Toc484596757)

[3.2 Proses Desain 15](#_Toc484596758)

[3.2.1 Perancangan Desain Sistem 15](#_Toc484596759)

[3.3.2 Perancangan Basis Data 32](#_Toc484596760)

[BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM 38](#_Toc484596761)

[4.1 Implementasi 38](#_Toc484596762)

[4.1.1 Implementasi Basis Data 38](#_Toc484596763)

[4.1.2 Implementasi Antarmuka Aplikasi JMN Anywhere pada Android 41](#_Toc484596764)

[4.1.3 Implementasi Antarmuka Aplikasi *Dashboard Admin* JMN Anywhere pada *Web* 53](#_Toc484596765)

[4.2 Pengujian 60](#_Toc484596766)

[4.2.1 Pengujian Aplikasi JMN Anywhere pada Android 61](#_Toc484596767)

[4.2.2 Pengujian Aplikasi *Dashboard Admin* JMN Anywhere pada *Web* 66](#_Toc484596768)

[4.3 Pembahasan 70](#_Toc484596769)

[BAB V PENUTUP 72](#_Toc484596770)

[5.1 Kesimpulan 72](#_Toc484596771)

[5.2 Saran 72](#_Toc484596772)

[DAFTAR PUSTAKA 73](#_Toc484596773)

# DAFTAR GAMBAR

[Gambar 2.1 Arsitektur *video streaming* 7](#_Toc483299736)

[Gambar 3.1 Diagram *use case* dari sistem aplikasi JMN Anywhere 17](#_Toc484594513)

[Gambar 3.2 Diagram aktivitas *login* pada aplikasi *web* 27](#_Toc484594514)

[Gambar 3.3 Diagram aktivitas *login* pada aplikasi Android 27](#_Toc484594515)

[Gambar 3.4 Diagram aktivitas kirim notifikasi 28](#_Toc484594516)

[Gambar 3.5 Diagram aktivitas mengakses konten pada aplikasi Android 29](#_Toc484594517)

[Gambar 3.6 Diagram aktivitas *upgrade to premium* 30](#_Toc484594518)

[Gambar 3.7 Diagram aktivitas pencarian konten pada aplikasi Android 30](#_Toc484594519)

[Gambar 3.8 Diagram aktivitas *edit profile* pada aplikasi Android 31](#_Toc484594520)

[Gambar 3.9 Diagram aktivitas *change password* pada aplikasi Android 31](#_Toc484594521)

[Gambar 3.10 *Deployment Diagram* aplikasi JMN Anywhere 32](#_Toc484594522)

[Gambar 3.11 Relasi antar entitas secara keseluruhan 36](#_Toc484594523)

[Gambar 4.1 Tampilan halaman *splash screen* aplikasi JMN Anywhere pada Android 42](#_Toc484594596)

[Gambar 4.2 Tampilan halaman *login* aplikasi JMN Anywhere pada Android 43](#_Toc484594597)

[Gambar 4.3 Tampilan halaman *forgot password* aplikasi JMN Anywhere pada Android 43](#_Toc484594598)

[Gambar 4.4 Tampilan halaman *register* aplikasi JMN Anywhere pada Android 44](#_Toc484594599)

[Gambar 4.5 Tampilan halaman *home free user* aplikasi JMN Anywhere pada Android 45](#_Toc484594600)

[Gambar 4.6 Tampilan halaman *home premium user* aplikasi JMN Anywhere pada Android 45](#_Toc484594601)

[Gambar 4.7 Tampilan halaman *live tv* aplikasi JMN Anywhere pada Android 46](#_Toc484594602)

[Gambar 4.8 Tampilan halaman *movies* aplikasi JMN Anywhere pada Android 47](#_Toc484594603)

[Gambar 4.9 Tampilan halaman *movie info* aplikasi JMN Anywhere pada Android 47](#_Toc484594604)

[Gambar 4.10 Tampilan halaman *profile free user* (kiri) dan *premium user* (kanan) aplikasi JMN Anywhere pada Android 48](#_Toc484594605)

[Gambar 4.11 Tampilan halaman *edit profile* aplikasi JMN Anywhere pada Android 49](#_Toc484594606)

[Gambar 4.12 Tampilan halaman *change password* aplikasi JMN Anywhere pada Android 49](#_Toc484594607)

[Gambar 4.13 Tampilan halaman *switch to premium* aplikasi JMN Anywhere pada Android 50](#_Toc484594608)

[Gambar 4.14 Tampilan halaman hasil pencarian aplikasi JMN Anywhere pada Android 51](#_Toc484594609)

[Gambar 4.15 Tampilan halaman *video player* aplikasi JMN Anywhere pada Android 51](#_Toc484594610)

[Gambar 4.16 Tampilan halaman *about* JMN Anywhere pada Android 52](#_Toc484594611)

[Gambar 4.17 Tampilan notifikasi tanpa gambar (kiri) dan dengan gambar (kanan) aplikasi JMN Anywhere pada Android 52](#_Toc484594612)

[Gambar 4.18 Tampilan halaman *login* aplikasi JMN Anywhere pada *web* 53](#_Toc484594613)

[Gambar 4.19 Tampilan halaman utama aplikasi JMN Anywhere pada *web* 54](#_Toc484594614)

[Gambar 4.20 Tampilan halaman kelola iklan aplikasi JMN Anywhere pada *web* 54](#_Toc484594615)

[Gambar 4.21 Tampilan halaman kelola *live tv* aplikasi JMN Anywhere pada *web* 55](#_Toc484594616)

[Gambar 4.22 Tampilan halaman kelola *movies* aplikasi JMN Anywhere pada *web* 56](#_Toc484594617)

[Gambar 4.23 Tampilan halaman detail *movies* aplikasi JMN Anywhere pada *web* 56](#_Toc484594618)

[Gambar 4.24 Tampilan halaman kelola *user* aplikasi JMN Anywhere pada *web* 57](#_Toc484594619)

[Gambar 4.25 Tampilan halaman *detail user* aplikasi JMN Anywhere pada *web* 57](#_Toc484594620)

[Gambar 4.26 Tampilan halaman pemberitahuan aplikasi JMN Anywhere pada *web* 58](#_Toc484594621)

[Gambar 4.27 Tampilan halaman kirim pemberitahuan aplikasi JMN Anywhere pada *web* 59](#_Toc484594622)

[Gambar 4.28 Tampilan halaman *reset password* aplikasi JMN Anywhere pada *web* 59](#_Toc484594623)

[Gambar 4.29 Tampilan halaman *go premium* aplikasi JMN Anywhere pada *web* 60](#_Toc484594624)

# DAFTAR TABEL

[Tabel 3.1 Definisi aktor diagram *use case* aplikasi JMNAnywhere 16](#_Toc484594770)

[Tabel 3.2 Skenario *use case* login 17](#_Toc484594771)

[Tabel 3.3 Skenario *use case* login 18](#_Toc484594772)

[Tabel 3.4 Skenario *use case* kelola iklan 18](#_Toc484594773)

[Tabel 3.5 Skenario *use case* kelola *live tv* 19](#_Toc484594774)

[Tabel 3.6 Skenario *use case* kelola *movies* 19](#_Toc484594775)

[Tabel 3.7 Skenario *use case* kelola *user* 20](#_Toc484594776)

[Tabel 3.8 Skenario *use case* kelola notifikasi 21](#_Toc484594777)

[Tabel 3.9 Skenario *use case* setujui permintaan akun premium 21](#_Toc484594778)

[Tabel 3.10 Skenario *use case* pencarian data 22](#_Toc484594779)

[Tabel 3.11 Skenario *use case* menonton *tv local* 22](#_Toc484594780)

[Tabel 3.12 Skenario *use case* menonton semua konten *live tv* 22](#_Toc484594781)

[Tabel 3.13 Skenario *use case* menonton *movies* 23](#_Toc484594782)

[Tabel 3.14 Skenario *use case* memilih kategori televisi dan *movies* 23](#_Toc484594783)

[Tabel 3.15 Skenario *use case* pencarian konten 24](#_Toc484594784)

[Tabel 3.16 Skenario *use case* *edit profil* 24](#_Toc484594785)

[Tabel 3.17 Skenario *use case edit password* 25](#_Toc484594786)

[Tabel 3.18 Skenario *use case* kirim permintaan *upgrade to premium* 25](#_Toc484594787)

[Tabel 3.19 Skenario *use case switch to premium account* 26](#_Toc484594788)

[Tabel 3.20 Skenario *use case logout* 26](#_Toc484594789)

[Tabel 3.21 Deskripsi entitas pada basis data 33](#_Toc484594790)

[Tabel 3.22 Rincian lengkap atribut dari entitas *genre* 33](#_Toc484594791)

[Tabel 3.23 Rincian lengkap atribut dari entitas *genre\_movies* 33](#_Toc484594792)

[Tabel 3.24 Rincian lengkap atribut dari entitas iklan 34](#_Toc484594793)

[Tabel 3.25 Rincian lengkap atribut dari entitas kategori\_*movie* 34](#_Toc484594794)

[Tabel 3.26 Rincian lengkap atribut dari entitas kategori\_tv 34](#_Toc484594795)

[Tabel 3.27 Rincian lengkap atribut dari entitas *level* 34](#_Toc484594796)

[Tabel 3.28 Rincian lengkap atribut dari entitas *live*\_tv 34](#_Toc484594797)

[Tabel 3.29 Rincian lengkap atribut dari entitas *movies* 35](#_Toc484594798)

[Tabel 3.30 Rincian lengkap atribut dari entitas notif 35](#_Toc484594799)

[Tabel 3.31 Rincian lengkap atribut dari entitas tbl\_kategori\_*movie* 35](#_Toc484594800)

[Tabel 3.32 Rincian lengkap atribut dari entitas *users* 35](#_Toc484594801)

[Tabel 4.1 Struktur tabel *users* 38](#_Toc484594914)

[Tabel 4.2 Struktur tabel *level* 38](#_Toc484594915)

[Tabel 4.3 Struktur tabel iklan 39](#_Toc484594916)

[Tabel 4.4 Struktur tabel *live\_tv* 39](#_Toc484594917)

[Tabel 4.5 Struktur tabel kategori\_tv 39](#_Toc484594918)

[Tabel 4.6 Struktur tabel *movies* 40](#_Toc484594919)

[Tabel 4.7 Struktur tabel *genre* 40](#_Toc484594920)

[Tabel 4.8 Struktur tabel *genre\_movies* 40](#_Toc484594921)

[Tabel 4.9 Struktur tabel kategori\_*movie* 40](#_Toc484594922)

[Tabel 4.10 Struktur tabel tbl\_kategori\_*movie* 41](#_Toc484594923)

[Tabel 4.11 Struktur tabel notif 41](#_Toc484594924)

[Tabel 4.12 Tabel pengujian pada telepon pintar berbasis Android 61](#_Toc484594925)

[Tabel 4.13 Tabel pengujian fungsi umum aplikasi Android 62](#_Toc484594926)

[Tabel 4.14 Tabel pengujian fungsi sisi *free user* 64](#_Toc484594927)

[Tabel 4.15 Tabel pengujian fungsi sisi *premium user* 65](#_Toc484594928)

[Tabel 4.16 Tabel pengujian fungsi umum pada *web* 67](#_Toc484594929)

[Tabel 4.17 Tabel pengujian fungsi pada menu kelola iklan 68](#_Toc484594930)

[Tabel 4.18 Tabel pengujian fungsi pada menu kelola *live tv* 68](#_Toc484594931)

[Tabel 4.19 Tabel pengujian fungsi pada menu kelola *movies* 69](#_Toc484594932)

[Tabel 4.20 Tabel pengujian fungsi pada menu kelola *users* 69](#_Toc484594933)

[Tabel 4.21 Tabel pengujian fungsi pada menu pemberitahuan 70](#_Toc484594934)

# BAB I PENDAHULUAN

## Latar Belakang

Di era moderen, banyak bermunculan inovasi-inovasi baru dalam dunia teknologi, seperti perkembangan teknologi media penyiaran televisi yang beralih dari televisi analog menuju televisi digital. Televisi digital yaitu media penyiaran dengan sistem transimisi digital melalui media satelit atau biasa disebut sebagai siaran televisi berlangganan. Hal ini memunculkan banyak perusahaan di dunia termasuk Indonesia yang menyediakan jasa penyiaran televisi berlangganan berbasis televisi satelit maupun televisi kabel yang memberikan siaran-siaran televisi dengan mencakup siaran di seluruh dunia dengan kualitas gambar *High Definiton* (HD).

PT SIMS Jogja Medianet merupakan salah satu perusahaan di Indonesia yang menyediakan jasa penyiaran televisi berlangganan berbasis televisi satelit maupun televisi kabel. Perusahaan ini masih menggunakan proses penyiaran pada umumnya yaitu dimulai ketika *provider* memancarkan siarannya ke satelit (*uplink*) kemudian sinyal tersebut ditransfer dan dikirim lagi menuju ke bumi (*downlink*) kemudian sinyal tersebut diterima oleh sebuah alat bernama *decoder* dan pelanggan dapat menikmatinya melalui pesawat televisi.

Akan tetapi, perkembangan perangkat bergerak seperti telepon pintar saat ini sangat banyak digunakan oleh masyarakat. Banyak aplikasi telepon pintar saat ini yang dibuat untuk memudahkan penggunanya untuk memenuhi kebutuhan sehari-hari.Seperti halnya untuk mendapatkan informasi-informasi yang terkini dari berbagai penjuru dunia. Hal itu membuat masyarakat mempunyai ketergantungan yang tinggi terhadap telepon pintar. Oleh karenanya, masyarakat jarang menonton siaran televisi secara langsung melalui televisi untuk mendapatkan informasi terkini karena terbantu oleh informasi yang didapatkan dengan mengakses internet melalui telepon pintar.

Banyak perusahaan-perusahaan penyedia televisi berlangganan di luar negeri yang sudah meluncurkan aplikasi *live video streamin*g berbasis Android yang

bertujuan untuk memudahkan para pelanggan mereka dalam mengakses siaran televisi berlangganan mereka, contohnya DISH yang merupakan perusahaan yang berpusat di Amerika Serikat. DISH menawarkan Hopper DVR dan aplikasi berbasis *mobile* yang memungkinkan seseorang tidak dapat hanya mengelola konten DVR tetapi juga melihat siaran langsung televisi dan konten DVR dimana saja yang terdapat koneksi internet berkecepatan tinggi, termasuk 3G. Dan seperti yang dikutip Streaming Media Industry Sourcebook, di tahun 2014 sebagai tahun bahwa MSOs (*Multiple Systems Operator*) tradisional akan menerapkan *streaming* dan kapan saja, dimana saja untuk mengakses ke DVR, VOD, dan siaran langsung televisi sekaligus. Mungkin mereka menyadari bahwa portal *video streaming* akan berkembang, dan mereka lebih baik dapat beralih atau mereka akan kehilangan pangsa pasar untuk generasi MSOs virtual berikutnya[1].

Berdasarkan hal tersebut, Penulis bermaksud merancang aplikasi *live video streaming* yang dapat digunakan sebagai media penyiaran untuk para pelanggan PT SIMS Jogja Medianet dalam menikmati layanan televisi berbayar mereka. Dengan aplikasi ini diharapkan para pelanggan PT SIMS Jogja Medianet merasa puas dengan layanan yang mereka tawarkan dan dapat menarik para pelanggan baru untuk berlangganan televisi berbayar.

## Rumusan Masalah

Bagaimana cara merancang sebuah aplikasi *live video streaming* berbasis Android yang dapat memberikan kemudahan dan kepuasan masyarakat khususnya pelanggan PT SIMS Jogja Medianet dalam menikmati layanan siaran langsung televisi berbayar melalui telepon pintar mereka?

## Batasan Masalah

Untuk menghindari pembahasan yang meluas, maka dalam Tugas Akhir ini ditetapkan batasan-batasan masalah sebagai berikut:

1. Aplikasi ini dibuat hanya untuk media penyiaran televisi dan *video on demand* melalui *video streaming.*
2. Aplikasi ini dibuat hanya untuk sisi *client* atau *user* yang terbagi menjadi dua yaitu akun *free user* dan akun *premium user.*
3. Hasil aplikasi *live video streaming* berbasis Android.
4. Hasil produk pengembangan bersifat *closed beta* yang belum dapat digunakan secara umum.
5. Hasil aplikasi *live video streaming* disimulasikan pada telepon pintar berbasis Android.

## Tujuan Penelitian

Tugas Akhir ini bertujuan untuk merancang sebuah aplikasi *live video streaming* berbasis Android yang dapat digunakan sebagai media penyiaran televisi melalui *video* *streaming* dengan koneksi internet atau *mobile data* sehingga diharapkan para pelanggan PT SIMS Jogja Medianet maupun pengguna lain dapat mengakses siaran televisi secara langsung dimana saja dan kapan saja melalui telepon pintar yang mereka miliki.

## Manfaat Penelitian

Manfaat yang diperoleh dari pembuatan aplikasi “JMN Anywhere” sebagai aplikasi *live video streaming* pada PT SIMS Jogja Medianet berbasis Android adalah sebagai berikut :

### Manfaat bagi Penulis

1. Memberikan pengalaman dalam melakukan perancangan perangkat lunak yang berdasarkan pada proses kerja dalam suatu perusahaan.
2. Mengimplementasikan ilmu yang dimiliki untuk memberikan solusi alternatif bagi PT SIMS Jogja Medianet untuk membuat layanan media penyiaran televisi melalui *video streaming* berbasis Android.

### Manfaat bagi pengguna perangkat lunak

Aplikasi *live video streaming* “JMN Anywhere” ini dapat memberikan kemudahan dan kenyamanan para pelanggan PT SIMS Jogja Medianet maupun pengguna lain dalam mengakses siaran televisi secara langsung dari telepon pintar berbasis Android.

## Sistematika Penulisan

Laporan Tugas Akhir ini disusun menurut sistematika penulisan yang terdiri atas lima bab dengan susunan sebagai berikut :

**BAB I PENDAHULUAN**

Bab ini berisi tentang latar belakang, rumusan masalah, pembatasan masalah, tujuan Tugas Akhir, manfaat penelitian, dan sistematika penulisan.

**BAB II LANDASAN TEORI**

Bab ini berisi uraian mengenai penelitian terdahulu, pengenalan *video streaming*, gambaran kerja *video streaming,* dan teknologi *video streaming.*

**BAB III PERANCANGAN SISTEM**

Bab ini berisi tentang perancangan aplikasi “JMN Anywhere” sebagai aplikasi *live video streaming* berbasis Android. Perancangan sistem berupa perancangan desain sistem menggunakan UML, perancangan basis data, dan analisis kebutuhan aplikasi.

**BAB IV HASIL DAN PEMBAHASAN**

Bab ini berisi tentang hasil dari perancangan aplikasi yang telah dibuat dan pengujian aplikasi *live video streaming* “JMN Anywhere” berbasis Android. Pengujian hasil aplikasi menggunakan metode pengujian *Black-Box*.

**BAB V PENUTUP**

Bab ini berisi tentang kesimpulan mengenai aplikasi yang telah dibangun apakah telah sesuai dengan perancangannya serta saran untuk perbaikan maupun pengembangan di masa yang akan datang.

# BAB II LANDASAN TEORI

## Penelitian Terdahulu

Sebelum melakukan perancangan sistem diperlukan beberapa studi literatur. Hemant Joshi berpendapat bahwa konsumsi media *online* telah menunjukkan pertumbuhan yang luar biasa selama beberapa tahun terakhir. Di antara perangkat digital, perangkat *mobile* telah mengambil alih sebagai media yang lebih disukai dari mengkonsumsi media *online*. Perangkat telepon pintar di seluruh dunia tumbuh pada tingkat pertumbuhan tahunan sebesar 17% dibandingkan dengan pertumbuhan 9,5% di semua perangkat *mobile*. Peningkatan jumlah perangkat *mobile* ini memudahkan konsumen untuk mengakses musik dan video konten di mana saja. Secara global, lalu lintas video dan audio telah mendominasi konsumsi internet data untuk beberapa tahun sekarang [2].

A. Albasir, dkk. [3] dalam penelitiannya menjelaskan bahwa telepon pintar adalah sarana favorit komunikasi antara orang-orang di seluruh dunia. Di antara semua jenis aplikasi telepon pintar, email, *streaming* multimedia, dan *browsing* web *mobile* adalah yang lebih populer. Pada dasarnya, *web* *browsing* dan *streaming* multimedia adalah aplikasi yang dirancang untuk digunakan dalam lingkungan komputasi *desktop* yang mana komputer yang terhubung ke internet melalui media kabel. Sekarang, baik aplikasi *web browsing* dan media *streaming* menggunakan sekitar 75% dari lalu lintas data *mobile*.

Franc Kozamernik dalam penelitiannya berpendapat bahwa konsep media *streaming* telah mengalami pertumbuhan yang mengesankan. Penggunaan teknologi *streaming* dalam pengiriman audio dan video melalui internet sekarang mencapai jutaan orang yang menggunakan komputer pribadi mereka. Media *streaming* menawarkan acara olahraga, musik, berita, hiburan dan konten *on-demand*. Dengan jaringan *broadband* yang digunakan di banyak negara, dan teknologi video / audio kompresi maju dengan cepat, kualitas layanan audio dan video melalui internet meningkat dengan cepat [4].

## *Video Streaming*

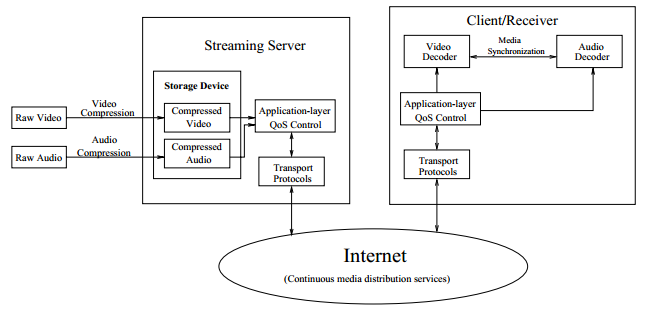
*Streaming* singkatan untuk *real-time* *transport live* media atau disimpan (misalnya video, audio dan data terkait) melalui Internet, antara komputer klien dan server. Ada dua mode untuk transmisi media melalui Internet, yaitu mode unduh dan mode *streaming*. Dalam mode unduh, pengguna dapat memutar *file* yang diunduh hanya setelah seluruh file telah diunduh dari server ke komputernya. Transfer file lengkap, dalam mode unduh, sering mengalami tidak dapat menerima waktu transfer lama, hal itu tergantung pada ukuran file media dan *bandwidth* dari saluran transportasi. Sedangkan dalam mode *streaming* file konten tidak perlu diunduh secara lengkap, tetapi memainkan sementara bagian dari konten yang diterima dan diterjemahkan.

*Video streaming* adalah satu-satunya teknologi yang mampu mengirimkan video dan audio melalui internet secara *real-time*. Selanjutnya, multimedia berkembang dari PC ke *platform* pengguna baru seperti perangkat genggam *nirkabel*, televisi interaktif kotak *set-top*, konsol *game*, telepon pintar, dan sebagainya[4].

## Gambaran Kerja *Video Streaming*

Menurut Dapeng Wu, dkk.[5], dalam *video streaming* terdapat enam bagian utama yaitu, kompresi video, layer aplikasi QoS kontrol, layanan distribusi media kontinyu, *streaming* server, mekanisme media sinkronisasi, dan protokol untuk media *streaming*. Hubungan antara enam blok bangunan dasar dapat diilustrasikan pada Gambar 2.1.

Gambar 2.1 menunjukkan arsitektur untuk *video streaming*. Pada Gambar 2.1, data video dan audio asli pra dikompresi dengan kompresi video dan algoritma kompresi audio dan kemudian disimpan dalam perangkat penyimpanan. Atas permintaan klien, server *streaming* mengambil video terkompresi atau data audio dari perangkat penyimpanan dan kemudian modul layer aplikasi QoS kontrol menyesuaikan video atau audio *bit-stream* sesuai dengan status jaringan dan persyaratan QoS. Setelah penyesuaian, protokol *transport packetize* terkompresi *bit-stream* dan mengirim paket video atau audio ke Internet. Paket mungkin mengalami keterlambatan berlebihan dalam Internet karena kemacetan transmisi. Untuk meningkatkan kualitas transmisi video atau audio, layanan distribusi media kontinyu (misalnya, *caching*) disebarkan di Internet. Untuk paket yang berhasil dikirim ke penerima, mereka pertama kali melewati layer *transport* dan kemudian diproses oleh layer aplikasi sebelum diterjemahkan di *decoder* video atau audio. Untuk mencapai sinkronisasi antara video dan presentasi audio, mekanisme sinkronisasi media diperlukan. Dari Gambar 2.1, dapat dilihat bahwa enam bagian yang erat terkait dan unsur yang jelas dari arsitektur *video streaming*[5].



Gambar 2.1 Arsitektur video streaming

Berikut penjelasan singkat dari enam bagian dari arsiktektur *video streaming*, yaitu :

1. *Video compression*

*Video compression* merupakan video asli harus dikompresi sebelum transmisi untuk mencapai efisiensi. Untuk *streaming* yang efektif, tingkat data konten harus lebih rendah daripada kecepatan koneksi pengguna, atau media tidak dapat ditonton[4].

1. *Application-layer QoS control*

*Application-layer QoS control* berguna untuk mengatasi kondisi jaringan yang bervariasi dan kualitas presentasi yang berbeda yang diminta oleh pengguna, berbagai teknik layer aplikasi QoS kontrol telah diusulkan. Teknik lapisan aplikasi termasuk kontrol kemacetan dan kontrol kesalahan. Kontrol kemacetan digunakan untuk mencegah *packet loss* dan mengurangi *delay*. Kontrol kesalahan, di sisi lain, adalah untuk meningkatkan kualitas presentasi video dengan adanya *packet loss*. Mekanisme kontrol kesalahan termasuk *foward error control* (FEC), transmisi ulang, *error-resilient* *encoding* dan penyembunyian kesalahan.

1. *Continuous media distribution services*

*Continuous media distribution services* agar dapat memberikan presentasi multimedia yang berkualitas, dukungan jaringan yang memadai sangat penting. Hal ini karena dukungan jaringan dapat mengurangi transportasi *delay* dan rasio *packet loss*. Layanan distribusi media kontinyu termasuk dalam *filtering* jaringan, *application-level multicast*, dan replikasi konten.

1. *Streaming servers*

Server *streaming* berperan penting dalam menyediakan layanan *streaming*. Untuk memberikan layanan *streaming* yang berkualitas, server *streaming* diminta untuk mengolah data multimedia yang di bawah batasan waktu dan mendukung operasi kontrol interaktif seperti *pause/resume*, *fast forward* dan *fast backward*. Selanjutnya, server *streaming* perlu untuk mengambil komponen media dalam mode sinkron. Sebuah server *streaming* biasanya terdiri dari tiga sub-sistem, yaitu, komunikator (misalnya, protokol transport), sistem operasi, dan sistem penyimpanan.

1. *Media synchronization mechanisms*

*Media synchronization mechanisms* merupakan fitur utama yang membedakan aplikasi multimedia dari aplikasi data tradisional lainnya. Media sinkronisasi mengacu pada menjaga hubungan waktu dalam satu aliran data dan antara berbagai media *stream*.

1. *Protocols for streaming media*

Protokol dirancang dan distandarisasi untuk komunikasi antara klien dan server *streaming*. Protokol untuk media *streaming* menyediakan layanan seperti jaringan pengalamatan, transportasi, dan *session control*. Berdasarkan fungsionalitas, protokol dapat diklasifikasikan dalam tiga kategori: protokol layer jaringan seperti *internet protocol* (IP), *transport protocol* seperti *user datagram protocol* (UDP), dan *session control* seperti *real-time streaming protocol* ( RTSP)[5].

## Teknologi *Video Streaming*

Dalam *video streaming* terdapat teknologi *streaming* yang berperan penting dalam proses pengiriman *video streaming* agar sampai ke pengguna atau *client.*

### Internet Protokol

Dalam distribusi *video streaming* dibutuhkan protokol, protokol dirancang dan distandarisasi untuk komunikasi antara klien dan server *streaming*. Beberapa protokol telah dikembangkan dan distandarisasi secara khusus untuk memungkinkan *streaming* konten multimedia secara *real-time* melalui Internet[4]. Berikut beberapa protokol yang dapat digunakan untuk *video streaming* :

1. UDP (*User Datagram Protocol*)

UDP merupakan metode yang disukai untuk pengiriman *live video streaming*. UDP adalah protokol yang lebih cepat daripada TCP (yaitu *live* video atau VoIP), protokol ini lebih baik digunakan untuk *live* video dengan kesalahan video yang disebabkan oleh paket yang terjatuh daripada menunggu pengiriman ulang yang menjamin TCP yang tidak terlalu praktis dimana *live* video yang bersangkutan[6].

1. TCP (*Transport Control Protocol*)

TCP membuat urutan *byte* yang ditransmisikan dan menginformasikan tujuan *byte* berikutnya yang tujuannya akan diterima. Jika *byte* tidak dikenali dalam jangka waktu tertentu, pesan tersebut akan dikirim ulang oleh sumbernya. Fitur ini memungkinkan perangkat mendeteksi dan mengidentifikasi paket yang hilang dan meminta pengiriman ulang.

1. HTTP (*HyperText Transfer Protocol*)

HTTP dikenal sebagai protokol yang secara efektif membawa halaman HTML dan memungkinkan *hyperlink* untuk mentransfer pengguna ke dokumen lain atau situs web. *Server* dan komputer klien memiliki koneksi dua arah, artinya ada umpan balik dari komputer klien (*receiver*). HTTP juga dapat digunakan untuk unduh media, terutama jika *file-*nya kecil dan jumlah pengguna bersamaan terbatas. Jika kecepatan koneksi lebih rendah dari kecepatan data media, media masih dapat melewatinya namun mungkin tidak berjalan lancar. Waktu transfer unduh file tergantung pada ukuran file dan kecepatan koneksi.

1. RTP (*Real-time Transport Protocol*)

Protokol transportasi ini dikembangkan secara khusus untuk mengalirkan data ke jaringan IP. RTP adalah standar *streaming* yang paling penting. Semua aliran media, terlepas dari format dan kontennya, dikemas dalam paket RTP. RTP berjalan di UDP dan menggunakan fungsi multipleks dan *checksum*-nya. Ini memungkinkan kontrol server media sehingga aliran video disajikan dengan kecepatan yang benar. *Media player* kemudian dapat memasang kembali paket RTP yang diterima ke dalam urutan yang benar dan memainkannya dengan kecepatan yang sesuai.

1. RTCP (*Real-Time Control Protocol*)

RTCP digunakan bersamaan dengan RTP dan menggunakan TCP untuk koneksi dua arah *client*-*server*. Ini memberikan umpan balik kepada penyedia layanan mengenai kualitas penerimaan jaringan dari masing-masing peserta dalam sesi RTP. Pesan tersebut mencakup laporan jumlah paket yang hilang dan statistik *jitter* (pendatang awal atau akhir). Sebagai contoh, *bit-rate* dari sebuah aliran dapat diubah untuk mengatasi kemacetan jaringan.

1. RTSP (*Real-Time Streaming Protocol*)

RTSP adalah protokol tingkat aplikasi untuk mengontrol data *real-time streaming*. RTSP menggunakan RTP sebagai protokol pengiriman data yang mendasarinya dan menawarkan VCR seperti kontrol kepada pengguna: *Play, Stop, Pause, FF* dan *REW*, serta akses acak ke bagian klip media manapun. RTSP juga membantu server menyesuaikan *bandwidth* media dengan kemacetan jaringan agar sesuai dengan kapasitas yang ada[4].

### *Server Stream Distribution*

Terdapat tiga metode dalam distribusi *server stream* yang digunakan yaitu *Unicast Transmission, Broadcast Transmission,* dan *Multicast Transmission.*

Berikut penjelasan dari ketiga metode tersebut [4]:

1. *Unicast Transmission*

Transmisi *Unicast* berarti mengirim satu stream ke masing-masing penerima. *Unicast* tidak mewakili penggunaan *bandwidth* yang sangat efisien namun memungkinkan pengguna, dengan menggunakan fungsionalitas RTSP, untuk menonton berbagai bagian media atau menonton film yang berbeda pada saat yang bersamaan. Penonton biasanya membuka media *unicast* dengan membuka URL RTSP.

1. *Broadcast Transmisson*

Transmisi *Broadcast* (dalam arti jaringan) berarti mengirim satu salinan *stream* ke seluruh jaringan. *Stream* tunggal dikirim ke semua host di jaringan. Kebanyakan LAN kecil mendukung *broadcast* tapi internet tidak mengizinkannya. *Broadcast* tidak mengizinkan penonton mengendalikan *stream* (yaitu tidak ada fungsi VCR). Tidak ada umpan balik dari pengguna ke server.

1. *Multicast Transmisson*

Transmisi *Multicast* berarti mengirim tepat satu salinan *stream*, tidak melalui keseluruhan jaringan (seperti dalam *Broadcast*) namun hanya di cabang-cabang jaringan dimana satu atau lebih penonton disesuaikan. Dengan cara ini, *bandwidth* jaringan yang tersedia dapat digunakan lebih banyak efisien. *Multicast* membutuhkan perangkat lunak *router* yang cukup canggih yang memungkinkan *server* membalas *stream* seperti yang diminta oleh klien.

# BAB III PERANCANGAN SISTEM



Tahapan yang digunakan dalam pembuatan sistem adalah model *waterfall*. Model *waterfall* merupakan model proses pengembangan perangkat lunak atau sering disebut siklus hidup (*life cycle*) klasik yang mana bersifat sistematis dengan menggunakan pendekatan sekuensial. Model *waterfall* mempunyai lima tahapan dalam proses pengembangan perangkat lunak yaitu Analisis Kebutuhan (*Requirements Analysis and Definition*)*,* Desain Sistem dan Perangkat Lunak(*System and Software Design*)*,* Implementasi(*Implementation and Unit Testing*)*,* Pengujian Sistem(*Integration and System Testing*)*,* dan Pemeliharaan(*Operation and Maintenance*)[7]*.*

Tahap analisis kebutuhan merupakan tahap awal dimana *programmer* berinteraksi dengan *costumer* untuk mengambil data tentang sistem pelayanan, kendala, dan tujuan yang ditetapkan pada saat bertemu dengan *costumer*. Tahap desain sistem merupakan proses perancangan sistem mengalokasikan kebutuhan baik perangkat keras atau perangkat lunak sistem dengan membentuk arsitektur sistem secara keseluruhan. Tahap implementasi atau pengkodean merupakan tahap dimana *programmer* mulai melakukan pengkodean atau *coding* dalam membangun sistem termasuk desain perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Tahap pengujian sistem merupakan tahap dimana unit program individu atau program dipadukan dan diuji sebagai sistem yang lengkap untuk memastikan bahwa kebutuhan perangkat lunak telah terpenuhi. Setelah pengujian, sistem perangkat lunak diserahkan kepada pengguna atau costumer. Tahap pemeliharaan merupakan fase siklus hidup terpanjang. Sistem terpasang dan digunakan secara nyata. Pemeliharaan meliputi mengoreksi kesalahan yang tidak ditemukan pada tahap-tahap sebelumnya, meningkatkan implementasi unit sistem dan meningkatkan layanan sistem sebagai kebutuhan baru[7].

## Analisis Kebutuhan

Dalam pengembangan perangkat lunak, terdapat beberapa kebutuhan sistem yang terdiri dari kebutuhan fungsional dan kebutuhan non-fungsional. Aplikasi *video streaming* ini berfungsi untuk media penyiaran televisi melalui telepon pintar yang dapat diakses dimanapun dan kapanpun. Sehingga, analisa kebutuhan diperlukan untuk mendapatkan hasil pengembangan perangkat lunak yang sesuai dengan kebutuhan-kebutuhan calon pengguna.

### Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang mendiskripsikan layanan, fitur atau fungsi yang disediakan atau diberikan oleh sistem yang nantinya digunakan oleh pengguna. Analisis kebutuhan fungsional sangat penting untuk mengetahui apakah kebutuhan dari pengguna sudah terpenuhi atau belum. Berikut merupakan kebutuhan fungsional dari aplikasi JMN Anywhere :

1. Sistem membagi pengguna ke dalam 3 *level* yaitu admin, *free user,* dan *premium user.*
2. Pengguna dengan *level* admin dapat mengelola data iklan, *live tv*, *movie*, *user*, mengirim notifikasi ke *user* dan menyetujui permintaan premium akun melalui aplikasi *web*.
3. Pengguna *free user* dan *premium user* hanya mengakses aplikasi Android saja.
4. Pengguna dengan status *free user* hanya dapat mengakses siaran televisi lokal.
5. Pengguna dengan status *free user* dapat mengirim permintaan *upgrade to premium* ke admin.
6. Dalam *upgrade* akun ke *premium user* tidak menyediakan fitur pembayaran.
7. Pengguna dengan status *premium user* dapat mengakses semua konten televisi dan *movies.*
8. Pengguna dengan status *premium user* dapat melakukan pencarian untuk semua konten yang ada.
9. Pengguna *free user* dan *premium user* dapat mengubah *profile*.
10. Pengguna *free user* dan *premium user* dapat mengganti kata sandi.
11. *Video player* tidak menyediakan fungsi *setting* kualitas video.

### Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah kebutuhan yang mendeskripsikan sekumpulan batasan, karakteristik dan properti pada sistem, baik dalam lingkungan pengembangan maupun operasional, atau atribut kualitas yang harus dipenuhi oleh sistem. Berikut adalah kebutuhan non-fungsional dari aplikasi JMN Anywhere :

1. Operasional
2. Sistem aplikasi web untuk *dashboard* admin dijalankan melalui *web browser* (Firefox, Chrome, Opera, dan Internet Explorer)
3. Sistem aplikasi Android JMN Anywhere dijalankan pada perangkat bergerak berbasis Android dengan versi Android *Jelly Bean* (4.2) sampai versi Android *Nougat* (7.1).
4. Keamanan
5. Sistem aplikasi JMN Anywhere pada Android memiliki fungsi *login* yang membedakan status pengguna.
6. Sistem aplikasi *web* untuk *dashboard* admin hanya memiliki fungsi *login* untuk satu *level* pengguna yaitu admin.
7. Kata sandi yang disimpan dalam *database* di-*hashing* menggunakan *bcrypt* (*Blowfish Crypt*).
8. Informasi
   1. Aplikasi Web
      1. Digunakan untuk menampilkan informasi apabila pengguna salah memasukkan *username* atau *password* pada saat *login* di aplikasi *web.*
      2. Digunakan untuk menampilkan informasi saat pengguna berhasil menambah, mengubah, menghapus data, dan mengirim notifikasi.
      3. Digunakan untuk menampilkan informasi saat *input form* belum diisi atau kosong.
   2. Aplikasi Android
9. Digunakan untuk menampilkan informasi apabila pengguna salah memasukkan *username* dan *password* pada saat *login* di aplikasi Android.
10. Digunakan untuk menampilkan informasi saat *input form* belum diisi atau kosong.
11. Digunakan untuk menampilkan informasi apabila pengguna belum melakukan verifikasi *email*.
12. Digunakan untuk menampilkan informasi penawaran *upgrade to premium* untuk *free user* saat membuka tab *profile*.
13. Digunakan untuk menampilkan informasi apabila pengguna *free user* mengklik konten selain kategori televisi lokal.
14. Digunakan untuk menampilkan informasi apabila pengguna *free user* berhasil melakukan permintaan *upgrade to premium* ke admin dan saat berhasil memasukkan kode *premium* untuk *switch to premium*.
15. Digunakan untuk menampilkan informasi apabila pengguna berhasil mengubah *profile* dan mengganti *password*.
16. Digunakan untuk menampilkan informasi apabila tidak ada koneksi internet dan tersambung ke internet.
17. Digunakan untuk menampilkan informasi apabila tidak ada data yang dicari pada saat melakukan pencarian data.
18. Diguakan untuk menampilkan informasi apabila tidak dapat menampilkan data dari *server.*

## Proses Desain

Proses desain pada pengembangan aplikasi JMN Anywhere terbagi menjadi dua bagian yaitu desain sistem dan basis data. Adapun penjelasan dari masing-masing bagian adalah sebagai berikut.

### Perancangan Desain Sistem

Perancangan sistem harus memperhatikan sasaran yang nantinya menjadi pengguna aplikasi ini, yaitu masyarakat yang utamanya pelanggan dari PT Jogja Medianet. Sasaran lain dari aplikasi ini adalah masyarakat luas untuk dapat menikmati siaran televisi dengan mudah melalui telepon pintaryang dapat diakses dimanapun dan kapanpun sehingga nantinya diharapkan dapat menarik perhatian masyarakat untuk berlangganan di PT Jogja Medianet. Berdasarkan perencanaan kebutuhan tersebut dapat digambarkan secara umum sistem yang dibuat menggunakan pemodelan UML atau *Unified Modeling Language,* UML merupakan alat untuk menentukan dan memvisualisasikan sistem perangkat lunak. Ini termasuk jenis diagram standar yang menggambarkan dan memetakan secara visual aplikasi komputer atau desain sistem *database* dan struktur.[8] UML memiliki tujuh diagram standar, lima diantaranya adalah *usecase diagram*, *class diagram*, *sequence diagram*, *deployment diagram,* dan *activity diagram*.[9] Berikut merupakan proses desain sistem aplikasi JMN Anywhere menggunakan UML.

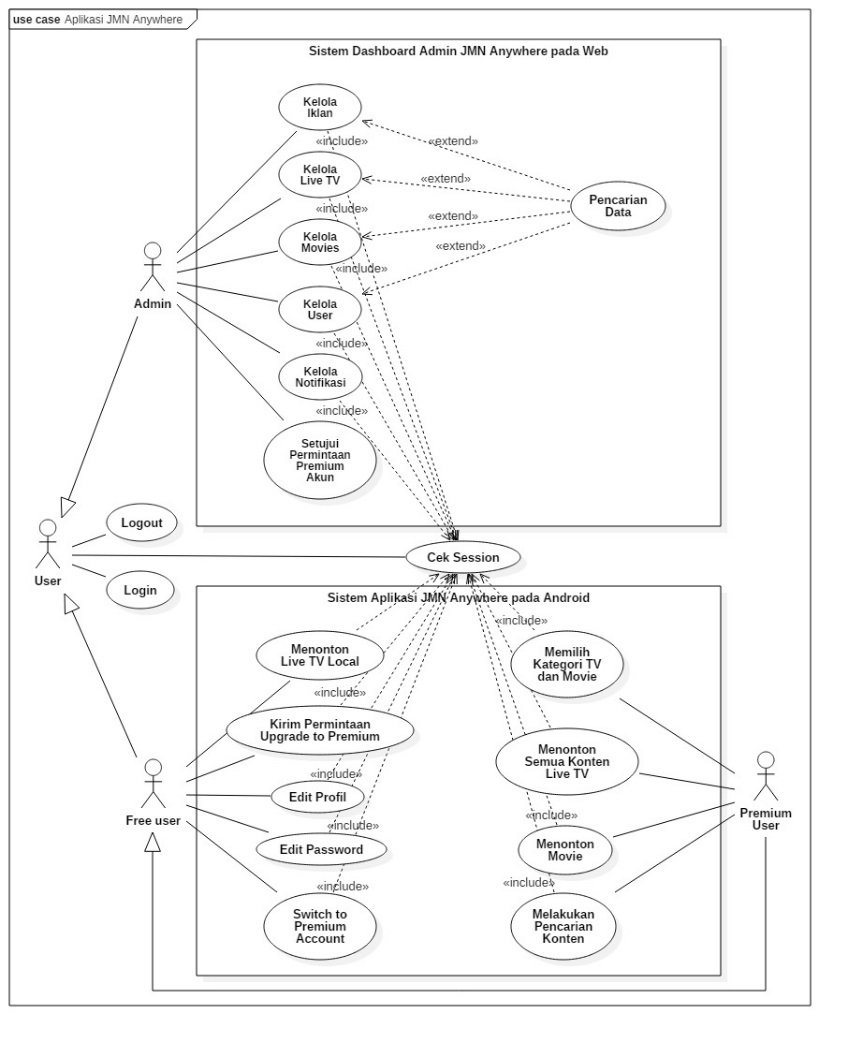
#### Diagram *Use Case*

Diagram *use case* digunakan untuk membantu tim pengembangan memvisualisasikan kebutuhan fungsional dari sistem, termasuk hubungan "aktor" (manusia yang akan berinteraksi dengan sistem) untuk proses penting, serta hubungan antara *use case* yang berbeda[9]. Pada tabel 3.1 berikut menunjukkan aktor yang terdapat dalam aplikasi JMN Anywhere.

Tabel 3.1 Definisi aktor diagram use case aplikasi JMNAnywhere

|  |  |
| --- | --- |
| **Aktor** | **Definisi** |
| *Administrator*/  Admin | Berperan sebagai *superuser,* admin dapat mengelola keseluruhan data yang ada di dalam aplikasi JMN Anywhere melalui halaman *web* yang meliputi kelola iklan, kelola *live tv*, kelola *movies*, kelola *user*, kelola notifikasi, dan setujui permintaan *premium* akun. |
| *Free user* | Aktor yang ada dalam sistem aplikasi JMN Anywhere pada Android dimana dapat mengakses siaran televisi lokal saja, mengirim permintaan *upgrade to premium*, mengubah data profil, dan mengganti *password* |
| *Premium user* | Aktor yang ada dalam sistem aplikasi JMN Anywhere pada Android dimana dapat mengakses semua konten yang ada di aplikasi JMN Anywhere, memilih kategori konten televisi dan *movie*, mengubah data profil, mengganti *password*, dan melakukan pencarian data untuk semua konten yang ada. |

Diagram *use case* dari aplikasi JMN Anyhere pada *web* dan Android ditunjukkan pada gambar 3.1.



Gambar 3.1 Diagram *use case* dari sistem aplikasi JMN Anywhere

Skenario *use case* berfungsi untuk memudahkan analisa pada tahap berikutnya dengan melakukan pendeskripsian *use case* secara lebih rinci seperti aktor yang berperan, tujuan *use case* tersebut dan kondisi awal saat *use case* tersebut akan dilakukan.

1. **Skenario *Use Case Login***

Pada tabel 3.2 menunjukkan skenario dari *use case* *login* dengan aktor yang terlibat adalah admin, *free user,* dan *premium user* dengan kondisi awal aktor telah membuka halaman *login*.

Tabel 3.2 Skenario use case login

|  |  |  |
| --- | --- | --- |
| **Identifikasi** | | |
| Nama *Use Case* | *Login* | |
| Aktor | Admin, *free user,* dan *premium user* | |
| Tujuan | Masuk ke dalam sistem | |
| Kondisi Awal | Membuka halaman *login* | |
| **Skenario** | | |
| **Aksi** | | **Respon Sistem** |
| 1. Mengisi *form username* dan *password* di halaman *login* | | 1. Melakukan autentikasi *username* dan *password* |
|  | | 1. Masuk ke dalam halaman utama sistem |
| **Skenario alternatif** | | |
| 1. Melakukan kesalahan pengisian *form username* dan *password* | | 1. Menampilkan pesan kesalahan dalam pengisian *form username* dan *password* |

1. **Skenario Use Case Cek Session**

Pada tabel 3.3 menunjukkan skenario dari *use case* cek *session* dengan aktor yang terlibat adalah admin, *free user,* dan *premium user* dengan kondisi awal aktor telah membuka halaman *login*.

Tabel 3.3 Skenario use case login

|  |  |  |
| --- | --- | --- |
| **Identifikasi** | | |
| Nama *Use Case* | Cek *Session* | |
| Aktor | Admin, *free user,* dan *premium user* | |
| Tujuan | Membedakan hak akses pengguna | |
| Kondisi Awal | Membuka halaman *login* | |
| **Skenario** | | |
| **Aksi** | | **Respon Sistem** |
| 1. Mengisi *form username* dan *password* di halaman *login* | | 1. Melakukan autentikasi *username* dan *password* |
|  | | 1. Melakukan pengecekan *level* hak akses pengguna |
|  | | 1. Masuk ke dalam halaman utama sistem |
| **Skenario alternatif** | | |
| 1. Melakukan kesalahan pengisian *form username* dan *password* | | 1. Menampilkan pesan kesalahan dalam pengisian *form username* dan *password* |

1. **Skenario Use Case Kelola Iklan**

Pada tabel 3.4 menunjukkan skenario dari *use case* kelola iklan dengan aktor yang terlibat adalah admin dengan kondisi awal aktor telah melakukan *login* ke dalam sistem pada aplikasi *web*.

Tabel 3.4 Skenario use case kelola iklan

|  |  |  |
| --- | --- | --- |
| **Identifikasi** | | |
| Nama *Use Case* | Kelola iklan | |
| Aktor | Admin | |
| Tujuan | Melihat, menambah, mengubah, atau menghapus data iklan | |
| Kondisi Awal | Aktor telah masuk ke dalam sistem pada aplikasi *web* | |
| **Skenario** | | |
| **Aksi** | | **Respon Sistem** |
|  | | 1. Menampilkan halaman utama sistem |
| 1. Aktor menekan menu data iklan dalam pilihan menu | | 1. Menampilkan halaman data iklan |
| 1. Menekan tombol tambah iklan baru | | 1. Menampilkan halaman tambah iklan baru |
| 1. Mengisi *form* tambah iklan baru dan menekan tombol simpan | | 1. Menyimpan data iklan pada basis data |
| 1. Menekan tombol edit pada salah satu data iklan | | 1. Menampilkan halaman edit data iklan |
| 1. Mengubah data iklan dan menekan tombol simpan | | 1. Menyimpan data pada basis data |
| 1. Menekan tombol hapus pada salah satu data iklan | | 1. Menampilkan dialog konfirmasi “Anda akan menghapus data iklan ini?” |
| 1. Menekan tombol ya pada dialog hapus | | 1. Menghapus data pada basis data jika aktor menekan ya kemudian kembali ke halaman data iklan. |
| 1. Menekan tombol tidak pada dialog hapus | | 1. Kembali ke halaman data iklan |

1. **Skenario *Use Case* Kelola *Live TV***

Pada tabel 3.5 menunjukkan skenario dari *use case* kelola *live tv* dengan aktor yang terlibat adalah admin dengan kondisi awal aktor telah melakukan *login* ke dalam sistem pada aplikasi *web*.

Tabel 3.5 Skenario use case kelola live tv

|  |  |  |
| --- | --- | --- |
| **Identifikasi** | | |
| Nama *Use Case* | Kelola *live tv* | |
| Aktor | Admin | |
| Tujuan | Melihat, menambah, mengubah, atau menghapus data *live tv* | |
| Kondisi Awal | Aktor telah masuk ke dalam sistem pada aplikasi *web* | |
| **Skenario** | | |
| **Aksi** | | **Respon Sistem** |
|  | | 1. Menampilkan halaman utama sistem |
| 1. Aktor menekan menu data *live tv* dalam pilihan menu | | 1. Menampilkan halaman data *live tv* |
| 1. Menekan tombol tambah *live tv* baru | | 1. Menampilkan halaman tambah *live tv* baru |
| 1. Mengisi *form* tambah *live tv* baru dan menekan tombol simpan | | 1. Menyimpan data *live tv* pada basis data |
| 1. Menekan tombol edit pada salah satu data *live tv* | | 1. Menampilkan halaman edit data *live tv* |
| 1. Mengubah data *live tv* dan menekan tombol simpan | | 1. Menyimpan data pada basis data |
| 1. Menekan tombol hapus pada salah satu data *live tv* | | 1. Menampilkan dialog konfirmasi “Anda akan menghapus data *live tv* ini?” |
| 1. Menekan tombol ya pada dialog hapus | | 1. Menghapus data pada basis data jika aktor menekan ya kemudian kembali ke halaman data *live tv*. |
| 1. Menekan tombol tidak pada dialog hapus | | 1. Kembali ke halaman data *live tv* |

1. **Skenario *Use Case* Kelola *Movies***

Pada tabel 3.6 menunjukkan skenario dari *use case* kelola *movies* dengan aktor yang terlibat adalah admin dengan kondisi awal aktor telah melakukan *login* ke dalam sistem pada aplikasi *web*.

Tabel 3.6 Skenario use case kelola movies

|  |  |  |
| --- | --- | --- |
| **Identifikasi** | | |
| Nama *Use Case* | Kelola *movies* | |
| Aktor | Admin | |
| Tujuan | Melihat, menambah, mengubah, atau menghapus data *movies* | |
| Kondisi Awal | Aktor telah masuk ke dalam sistem pada aplikasi *web* | |
| **Skenario** | | |
| **Aksi** | | **Respon Sistem** |
|  | | 1. Menampilkan halaman utama sistem |
| 1. Aktor menekan menu data *movies* dalam pilihan menu | | 1. Menampilkan halaman data *movies* |
| 1. Menekan tombol tambah *movies* baru | | 1. Menampilkan halaman tambah *movies* baru |
| 1. Mengisi *form* tambah *movies* baru dan menekan tombol simpan | | 1. Menyimpan data *movies* pada basis data |
| 1. Menekan tombol edit pada salah satu data *movies* | | 1. Menampilkan halaman edit data *movies* |
| 1. Mengubah data *movies* dan menekan tombol simpan | | 1. Menyimpan data pada basis data |
| 1. Menekan tombol hapus pada salah satu data *movies* | | 1. Menampilkan dialog konfirmasi “Anda akan menghapus data *movies* ini?” |
| 1. Menekan tombol ya pada dialog hapus | | 1. Menghapus data pada basis data jika aktor menekan ya kemudian kembali ke halaman data *movies*. |
| 1. Menekan tombol tidak pada dialog hapus | | 1. Kembali ke halaman data *movies* |

1. **Skenario *Use Case* Kelola *User***

Pada tabel 3.7 menunjukkan skenario dari *use case* kelola *user* dengan aktor yang terlibat adalah admin dengan kondisi awal aktor telah melakukan *login* ke dalam sistem pada aplikasi *web*.

Tabel 3.7 Skenario use case kelola user

|  |  |  |
| --- | --- | --- |
| **Identifikasi** | | |
| Nama *Use Case* | Kelola *user* | |
| Aktor | Admin | |
| Tujuan | Melihat atau menghapus data *user* | |
| Kondisi Awal | Aktor telah masuk ke dalam sistem pada aplikasi *web* | |
| **Skenario** | | |
| **Aksi** | | **Respon Sistem** |
|  | | 1. Menampilkan halaman utama sistem |
| 1. Aktor menekan menu data *user* dalam pilihan menu | | 1. Menampilkan halaman data *user* |
| 1. Menekan tombol hapus pada salah satu data *user* | | 1. Menampilkan dialog konfirmasi “Anda akan menghapus data *users* ini?” |
| 1. Menekan tombol ya pada dialog hapus | | 1. Menghapus data pada basis data jika aktor menekan ya kemudian kembali ke halaman data *user* |
| 1. Menekan tombol tidak pada dialog hapus | | 1. Kembali ke halaman data *user* |

1. **Skenario *Use Case* Kelola Notifikasi**

Pada tabel 3.8 menunjukkan skenario dari *use case* kelola notifikasi dengan aktor yang terlibat adalah admin dengan kondisi awal aktor telah melakukan *login* ke dalam sistem pada aplikasi *web*.

Tabel 3.8 Skenario use case kelola notifikasi

|  |  |  |
| --- | --- | --- |
| **Identifikasi** | | |
| Nama *Use Case* | Kelola notifikasi | |
| Aktor | Admin | |
| Tujuan | Mengirim atau menghapus notifikasi | |
| Kondisi Awal | Aktor telah masuk ke dalam sistem pada aplikasi *web* | |
| **Skenario** | | |
| **Aksi** | | **Respon Sistem** |
|  | | 1. Menampilkan halaman utama sistem |
| 1. Aktor menekan menu pemberitahuan dalam pilihan menu | | 1. Menampilkan halaman pemberitahuan |
| 1. Menekan tombol kirim pemberitahuan | | 1. Menampilkan *form* kirim pemberitahuan |
| 1. Mengisi *form* kirim pemberitahuan dan menekan tombol “Kirim” | | 1. Menyimpan data pemberitahuan pada basis data dan mengirim pemberitahuan ke aplikasi Android |
| 1. Menekan tombol hapus pada salah satu data pemberitahuan | | 1. Menampilkan dialog konfirmasi “Anda akan menghapus data pemberitahuan ini?” |
| 1. Menekan tombol ya pada dialog hapus | | 1. Menghapus data pada basis data jika aktor menekan ya kemudian kembali ke halaman pemberitahuan |
| 1. Menekan tombol tidak pada dialog hapus | | 1. Kembali ke halaman pemberitahuan |

1. **Skenario *Use Case* Setujui Permintaan Akun *Premium***

Pada tabel 3.9 menunjukkan skenario dari *use case* setujui permintaan akun *premium* dengan aktor yang terlibat adalah admin dengan kondisi awal aktor telah masuk ke *link form* konfirmasi persetujuan *upgrade to premium* pada aplikasi *web*.

Tabel 3.9 Skenario use case setujui permintaan akun premium

|  |  |  |
| --- | --- | --- |
| **Identifikasi** | | |
| Nama *Use Case* | Setujui permintaan akun *premium* | |
| Aktor | Admin | |
| Tujuan | Mengkonfirmasi persetujuan permintaan *upgrade to premium* dari akun *free user* | |
| Kondisi Awal | Aktor telah masuk ke *link* *form* konfirmasi persetujuan pada aplikasi *web* | |
| **Skenario** | | |
| **Aksi** | | **Respon Sistem** |
|  | | 1. Menampilkan halaman *form* konfirmasi persetujuan dengan mengambil data *user* dari basis data |
| 1. Mengecek data *free user* yang akan dikonfirmasi dan menekan tombol “*Accept*” | | 1. Menyimpan data pada basis data dan mengirim kode *premium* ke alamat email *free user* |

1. **Skenario *Use Case* Pencarian Data**

Pada tabel 3.10 menunjukkan skenario dari *use case* pencarian data dengan aktor yang terlibat adalah admin dengan kondisi awal aktor telah melakukan *login* ke dalam sistem pada aplikasi *web.*

Tabel 3.10 Skenario use case pencarian data

|  |  |  |
| --- | --- | --- |
| **Identifikasi** | | |
| Nama *Use Case* | Pencarian data | |
| Aktor | Admin | |
| Tujuan | Mencari data iklan, *live tv, movies, users*, dan pemberitahuan | |
| Kondisi Awal | Aktor telah masuk ke dalam sistem aplikasi *web* | |
| **Skenario** | | |
| **Aksi** | | **Respon Sistem** |
| 1. Memasukkan kata kunci pada *form* pencarian yang ada di dalam menu dan menekan *icon* *search* | | 1. Melakukan pencarian data yang tersimpan pada basis data sesuai dengan *keyword* dan menampilkannya pada halaman menu |

1. **Skenario *Use Case* Menonton *Live TV Local***

Pada tabel 3.11 menunjukkan skenario dari *use case* menonton *live tv local* dengan aktor yang terlibat adalah *free user* dan *premium user* dengan kondisi awal aktor telah melakukan *login* ke dalam sistem pada aplikasi Android*.*

Tabel 3.11 Skenario use case menonton tv local

|  |  |  |
| --- | --- | --- |
| **Identifikasi** | | |
| Nama *Use Case* | Menonton *live tv local* | |
| Aktor | *Free user* dan *premium user* | |
| Tujuan | Menonton siaran televisi dengan kategori televisi lokal | |
| Kondisi Awal | Aktor telah masuk ke dalam sistem aplikasi Android | |
| **Skenario** | | |
| **Aksi** | | **Respon Sistem** |
|  | | 1. Menampilkan halaman utama sistem atau *Home* |
| 1. Memilih tab *Home* atau *Live TV* | | 1. Menampilkan daftar stasiun televisi dengan kategori lokal di dalam menu tab *Home* dan *Live TV* |
| 1. Memilih dan mengklik stasiun televisi yang dipilih pada daftar televisi yang ditampilkan di menu tab *Home* atau *Live TV* | | 1. Memutar siaran sesuai stasiun televisi yang dipilih oleh aktor |

1. **Skenario *Use Case* Menonton Semua Konten *Live TV***

Pada tabel 3.12 menunjukkan skenario dari *use case* menonton semua konten *live tv* dengan aktor yang terlibat adalah *premium user* dengan kondisi awal aktor telah melakukan *login* ke dalam sistem pada aplikasi Android*.*

Tabel 3.12 Skenario use case menonton semua konten live tv

|  |  |  |
| --- | --- | --- |
| **Identifikasi** | | |
| Nama *Use Case* | Menonton semua konten *live tv* | |
| Aktor | *Premium user* | |
| Tujuan | Menonton semua siaran televisi yang ada | |
| Kondisi Awal | Aktor telah masuk ke dalam sistem aplikasi Android | |
| **Skenario** | | |
| **Aksi** | | **Respon Sistem** |
|  | | 1. Menampilkan halaman utama sistem atau *Home* |
| 1. Memilih tab *Home* atau *Live TV* | | 1. Menampilkan beberapa televisi di tab *Home* dan menampilkan semua daftar stasiun televisi yang ada di tab *Live TV* |
| 1. Memilih dan mengklik stasiun televisi yang dipilih pada daftar televisi yang ditampilkan di menu tab *Home* atau *Live TV* | | 1. Memutar siaran sesuai stasiun televisi yang dipilih oleh aktor |

1. **Skenario *Use Case* Menonton *Movies***

Pada tabel 3.13 menunjukkan skenario dari *use case* menonton movies dengan aktor yang terlibat adalah *premium user* dengan kondisi awal aktor telah melakukan *login* ke dalam sistem pada aplikasi Android*.*

Tabel 3.13 Skenario use case menonton movies

|  |  |  |
| --- | --- | --- |
| **Identifikasi** | | |
| Nama *Use Case* | Menonton *movies* | |
| Aktor | *Premium user* | |
| Tujuan | Menonton *movies* yang tersedia di tab *Movies* | |
| Kondisi Awal | Aktor telah masuk ke dalam sistem aplikasi Android | |
| **Skenario** | | |
| **Aksi** | | **Respon Sistem** |
|  | | 1. Menampilkan halaman utama sistem atau *Home* |
| 1. Memilih tab *Movies* | | 1. Menampilkan semua daftar pilihan *movies* yang ada di tab *Movies* |
| 1. Memilih dan mengklik *movies* yang dipilih pada daftar pilihan *movies* yang ditampilkan di menu tab *Movies* | | 1. Memutar *movies* sesuai yang dipilih oleh aktor |

1. **Skenario *Use Case* Memilih Kategori Televisi dan *Movies***

Pada tabel 3.14 menunjukkan skenario dari *use case* memilih kategori televisi dan *movies* dengan aktor yang terlibat adalah *premium user* dengan kondisi awal aktor telah melakukan *login* ke dalam sistem pada aplikasi Android*.*

Tabel 3.14 Skenario use case memilih kategori televisi dan movies

|  |  |  |
| --- | --- | --- |
| **Identifikasi** | | |
| Nama *Use Case* | Memilih kategori televisi dan *movies* | |
| Aktor | *Premium user* | |
| Tujuan | Menonton siaran televisi atau *movies* sesuai dengan kategori yang diinginkan | |
| Kondisi Awal | Aktor telah masuk ke dalam sistem aplikasi Android | |
| **Skenario** | | |
| **Aksi** | | **Respon Sistem** |
|  | | 1. Menampilkan halaman utama sistem atau *Home* |
| 1. Memilih tab *Live TV atau* *Movies* | | 1. Menampilkan daftar pilihan kategori siaran televisi yang ada di tab *Live TV* |
| 1. Memilih dan mengklik pada salah satu kategori yang dipilih pada daftar pilihan kategori yang ditampilkan di menu tab *Live TV* atau *Movies* | | 1. Menampilkan siaran televisi atau *movies* sesuai dengan kategori yang dipilih aktor |

1. **Skenario *Use Case* Pencarian Konten**

Pada tabel 3.15 menunjukkan skenario dari *use case* pencarian konten dengan aktor yang terlibat adalah *premium user* dengan kondisi awal aktor telah melakukan *login* ke dalam sistem pada aplikasi Android*.*

Tabel 3.15 Skenario use case pencarian konten

|  |  |  |
| --- | --- | --- |
| **Identifikasi** | | |
| Nama *Use Case* | Pencarian konten | |
| Aktor | *Premium user* | |
| Tujuan | Menemukan konten siaran televisi atau *movies* | |
| Kondisi Awal | Aktor telah masuk ke dalam sistem aplikasi Android | |
| **Skenario** | | |
| **Aksi** | | **Respon Sistem** |
|  | | 1. Menampilkan halaman utama sistem atau *Home* |
| 1. Mengklik ikon *search* pada *toolbar* | | 1. Menampilkan *form* pencarian pada *toolbar* |
| 1. Mengisi *form* pencarian dengan kata kunci pencarian dan menekan tombol enter | | 1. Menampilkan hasil pencarian data yang ada pada basis data sesuai dengan kata kunci yang diinginkan aktor |

1. **Skenario *Use Case Edit* Profil**

Pada tabel 3.16 menunjukkan skenario dari *use case edit* profil dengan aktor yang terlibat adalah *free user* dan *premium user* dengan kondisi awal aktor telah melakukan *login* ke dalam sistem pada aplikasi Android*.*

Tabel 3.16 Skenario use case edit profil

|  |  |  |
| --- | --- | --- |
| **Identifikasi** | | |
| Nama *Use Case* | *Edit* profil | |
| Aktor | *Free user* dan *premium user* | |
| Tujuan | Mengubah data profil dari pengguna | |
| Kondisi Awal | Aktor telah masuk ke dalam sistem aplikasi Android | |
| **Skenario** | | |
| **Aksi** | | **Respon Sistem** |
|  | | 1. Menampilkan halaman utama sistem atau *Home* |
| 1. Memilih tab *Profile* | | 1. Menampilkan data profil pengguna |
| 1. Menekan tombol “*Edit Profile*” | | 1. Menampilkan *form* *edit* profil |
| 1. Mengubah isi dari *form* *edit* profil yang ditampilkan dan menklik tombol *checklist* pada toolbar | | 1. Menyimpan perubahan data pada basis data dan kembali ke tab *Profile* |

1. **Skenario *Use Case Edit Password***

Pada tabel 3.17 menunjukkan skenario dari *use case edit* *password* dengan aktor yang terlibat adalah *free user* dan *premium user* dengan kondisi awal aktor telah melakukan *login* ke dalam sistem pada aplikasi Android*.*

Tabel 3.17 Skenario use case edit password

|  |  |  |
| --- | --- | --- |
| **Identifikasi** | | |
| Nama *Use Case* | *Edit password* | |
| Aktor | *Free user* dan *premium user* | |
| Tujuan | Mengubah kata sandi dari pengguna | |
| Kondisi Awal | Aktor telah masuk ke dalam sistem aplikasi Android | |
| **Skenario** | | |
| **Aksi** | | **Respon Sistem** |
|  | | 1. Menampilkan halaman utama sistem atau *Home* |
| 1. Memilih tab *Profile* | | 1. Menampilkan data profil pengguna |
| 1. Menekan tombol “*Change Password*” | | 1. Menampilkan *form* *change password* |
| 1. Mengisi *form* *change password* dengan memasukkan kata sandi lama dan kata sandi baru dan menklik tombol *checklist* pada toolbar | | 1. Melakukan autentikasi kata sandi lama, kemudian menyimpan kata sandi baru pada basis data dan kembali ke tab *Profile* |

1. **Skenario *Use Case* Kirim Permintaan *Upgrade to Premium***

Pada tabel 3.18 menunjukkan skenario dari *use case* kirim permintaan *upgrade to premium* dengan aktor yang terlibat adalah *free user* dengan kondisi awal aktor telah melakukan *login* ke dalam sistem pada aplikasi Android*.*

Tabel 3.18 Skenario use case kirim permintaan upgrade to premium

|  |  |  |
| --- | --- | --- |
| **Identifikasi** | | |
| Nama *Use Case* | Kirim permintaan *upgrade to premium* | |
| Aktor | *Free user* | |
| Tujuan | Mengirim permintaan *upgrade to premium* ke admin | |
| Kondisi Awal | Aktor telah masuk ke dalam sistem aplikasi Android | |
| **Skenario** | | |
| **Aksi** | | **Respon Sistem** |
|  | | 1. Menampilkan halaman utama sistem atau *Home* |
| 1. Memilih tab *Profile* | | 1. Menampilkan *snackbar* “*Upgrade to Premium*” |
| 1. Menekan tombol “*Go Premium*” | | 1. Menampilkan *form* *enter password* |
| 1. Mengisi *form* *enter password* dengan memasukkan kata sandi pengguna dan menekan tombol OK | | 1. Melakukan autentikasi *password*, dan mengirim permintaan *upgrade to premium* ke *email* admin |

1. **Skenario *Use Case* *Switch to Premium Account***

Pada tabel 3.19 menunjukkan skenario dari *use case switch to premium account* dengan aktor yang terlibat adalah *free user* dengan kondisi awal aktor telah melakukan *login* ke dalam sistem pada aplikasi Android*.*

Tabel 3.19 Skenario use case switch to premium account

|  |  |  |
| --- | --- | --- |
| **Identifikasi** | | |
| Nama *Use Case* | *Switch to premium account* | |
| Aktor | *Free user* | |
| Tujuan | Meng-*upgrade* akun dari *free user* ke *premium user* | |
| Kondisi Awal | Aktor telah masuk ke dalam sistem aplikasi Android | |
| **Skenario** | | |
| **Aksi** | | **Respon Sistem** |
|  | | 1. Menampilkan halaman utama sistem atau *Home* |
| 1. Memilih tab *Profile* | | 1. Menampilkan data *profile* |
| 1. Menekan tombol “*Switch to Premium*” | | 1. Menampilkan *form* *switch to premium* |
| 1. Memasukkan kode *premium* yang telah didapat ke *form* *switch to premium* dan menekan tombol OK | | 1. Melakukan validasi kode *premium* dengan yang tersimpan dalam basis data |

1. **Skenario *Use Case* *Logout***

Pada tabel 3.20 menunjukkan skenario dari *use case logout* dengan aktor yang terlibat adalah admin, *free user* dan *premium user* dengan kondisi awal aktor telah melakukan *login* ke dalam sistem*.*

Tabel 3.20 Skenario use case logout

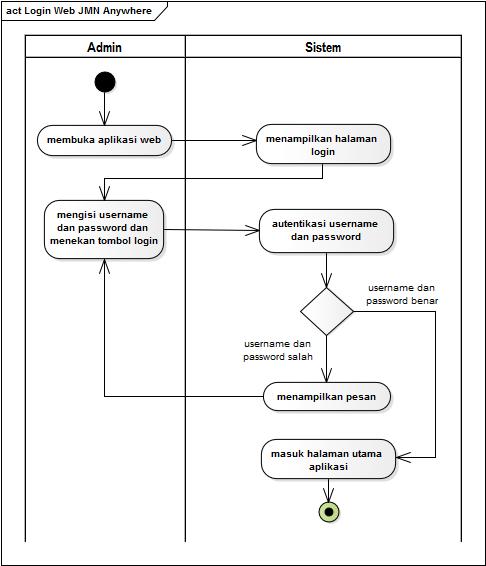
|  |  |  |
| --- | --- | --- |
| **Identifikasi** | | |
| Nama *Use Case* | *Logout* | |
| Aktor | Admin, *free user* dan *premium user* | |
| Tujuan | Keluar dari dalam sistem | |
| Kondisi Awal | Aktor telah masuk ke dalam sistem aplikasi | |
| **Skenario** | | |
| **Aksi** | | **Respon Sistem** |
|  | | 1. Menampilkan halaman utama sistem |
| 1. Mengklik tombol keluar atau *sign-out* | | 1. Menampilkan halaman *login* |

#### Diagram Aktivitas

Diagram aktivitas menunjukkan aliran prosedural kontrol antara dua atau lebih objek kelas saat memproses suatu aktivitas.[9] Dalam UML, diagram aktivitas dapat digunakan untuk menggambarkan kegiatan dan operasional langkah demi langkah alur kerja komponen dalam sistem. Diagram aktivitasmenunjukkan aliran keseluruhan kontrol.[8] Berikut adalah diagram aktivitas dari aplikasi JMN Anywhere.

1. **Diagram Aktivitas *Login* (*Web*)**

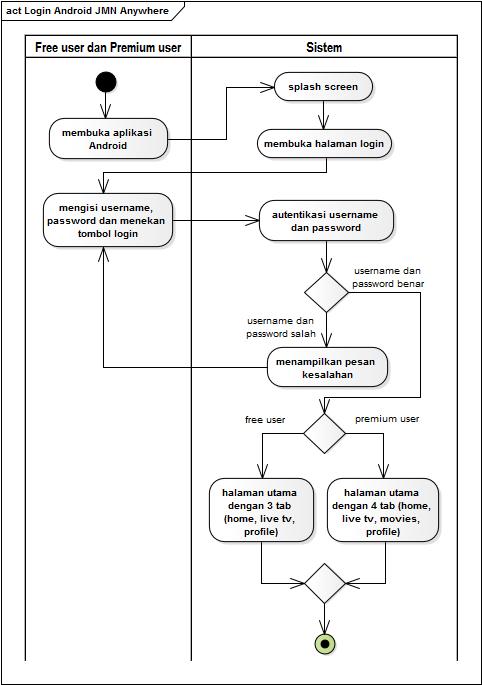
Pada gambar 3.2 menunjukkan diagram aktivitas dari *login* pada aplikasi *web*. Urutan aktivitas dimulai dari membuka aplikasi dan berakhir dengan *login* berhasil dan masuk ke halaman utama sistem.



Gambar 3.2Diagram aktivitas login pada aplikasi web

1. **Diagram Aktivitas *Login* (Android)**

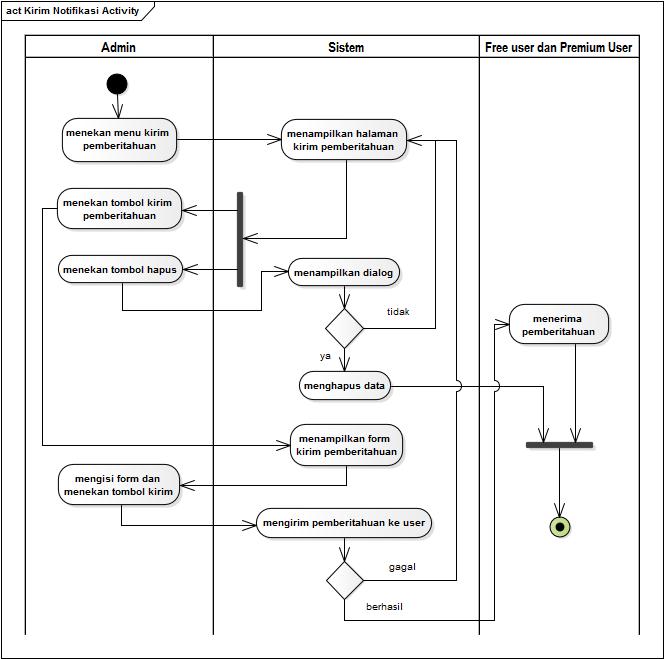
Pada gambar 3.3 menunjukkan diagram aktivitas dari *login* pada aplikasi Android. Urutan aktivitas dimulai dari membuka aplikasi dan berakhir dengan *login* berhasil dan masuk ke halaman utama sistem.



Gambar 3.3 Diagram aktivitas login pada aplikasi Android

1. **Diagram Aktivitas Kirim Notifikasi**

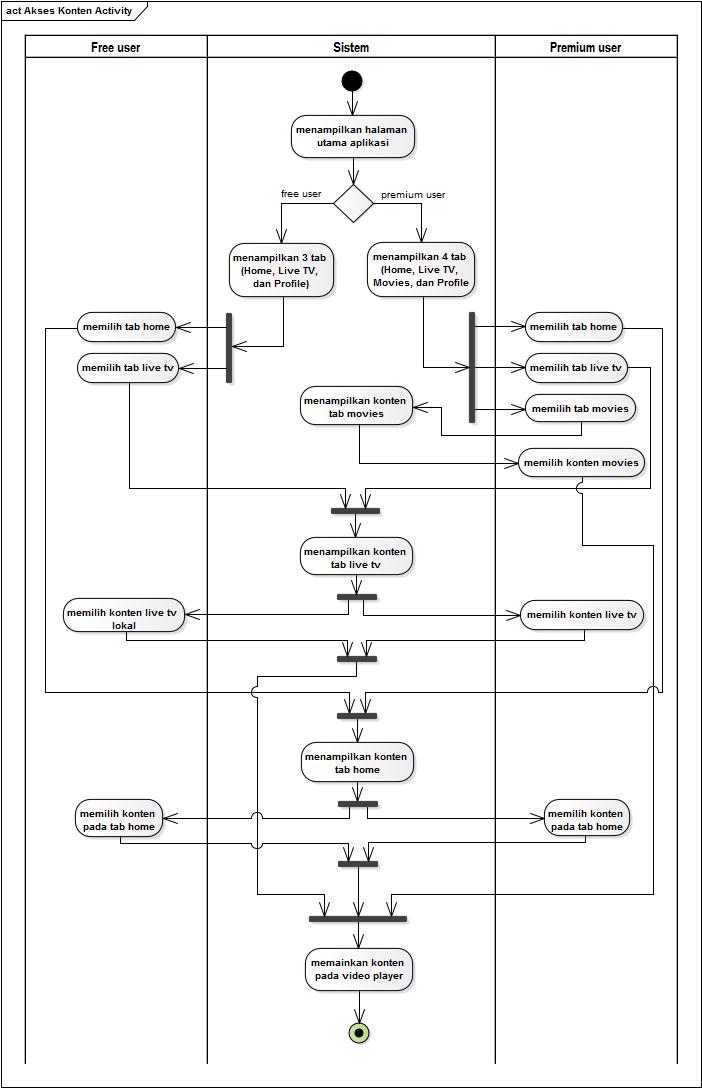
Pada gambar 3.4 menunjukkan diagram aktivitas dari kirim notifikasi. Urutan aktivitas dimulai dari membuka menu kirim notifikasi dan berakhir dengan pengguna menerima notifikasi pada aplikasi Android.

****

Gambar 3.4 Diagram aktivitas kirim notifikasi

1. **Diagram Aktivitas Mengakses Konten (Android)**

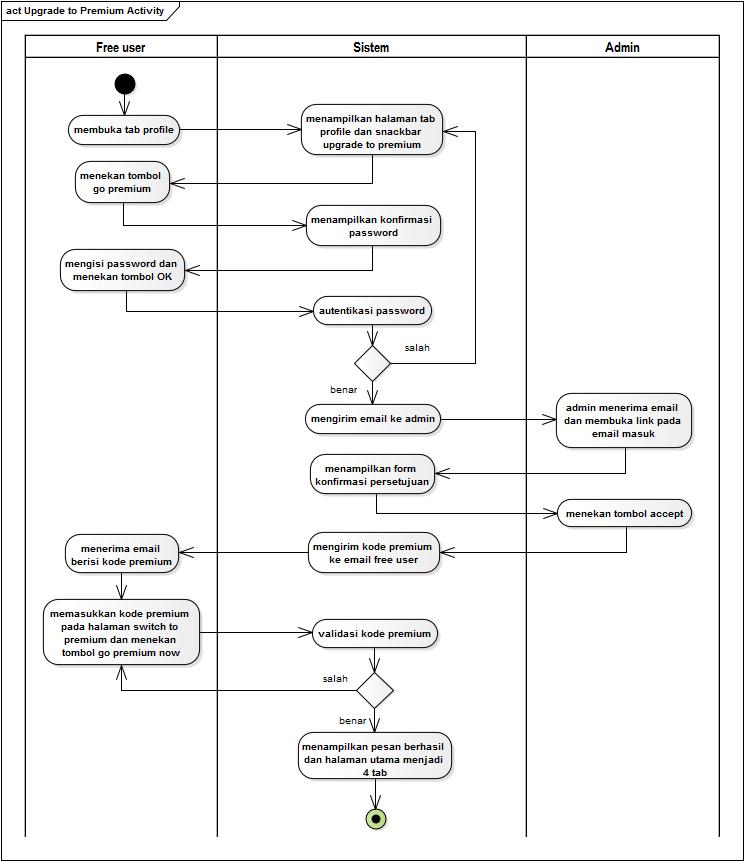
Pada gambar 3.5 menunjukkan diagram aktivitas dari mengakses konten pada aplikasiAndroid. Urutan aktivitas dimulai dari membuka halaman utama dari aplikasi dan berakhir dengan memainkan konten pada *video player*.



Gambar 3.5 Diagram aktivitas mengakses konten pada aplikasi Android

1. **Diagram Aktivitas *Upgrade to Premium***

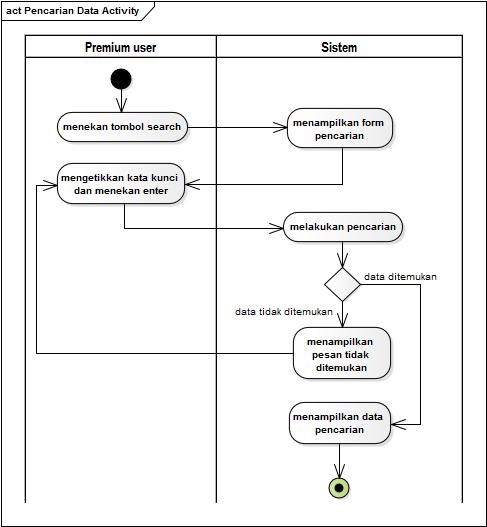
Pada gambar 3.6 menunjukkan diagram aktivitas dari *upgrade to premium*. Urutan aktivitas dimulai dari membuka halaman utama dari aplikasi dan berakhir dengan mengubah tampilan halaman utama dari *free user* ke *premium user*.



Gambar 3.6 Diagram aktivitas upgrade to premium

1. **Diagram Aktivitas Pencarian Konten (Android)**

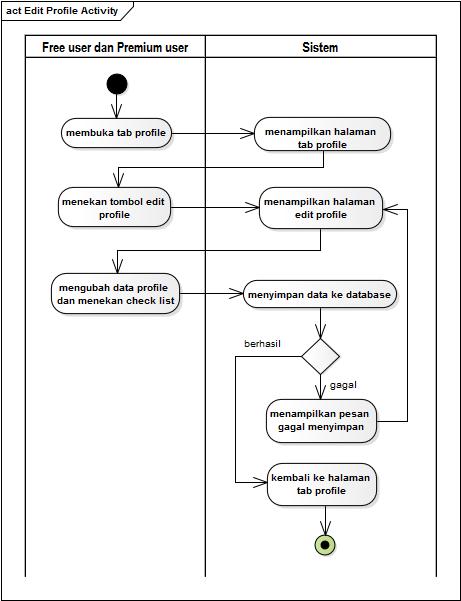
Pada gambar 3.7 menunjukkan diagram aktivitas dari pencarian konten pada aplikasiAndroid. Urutan aktivitas dimulai dari membuka halaman utama aplikasi dan berakhir dengan menampilkan data pencarian.



Gambar 3.7 Diagram aktivitas pencarian konten pada aplikasi Android

1. **Diagram Aktivitas *Edit Profile* (Android)**

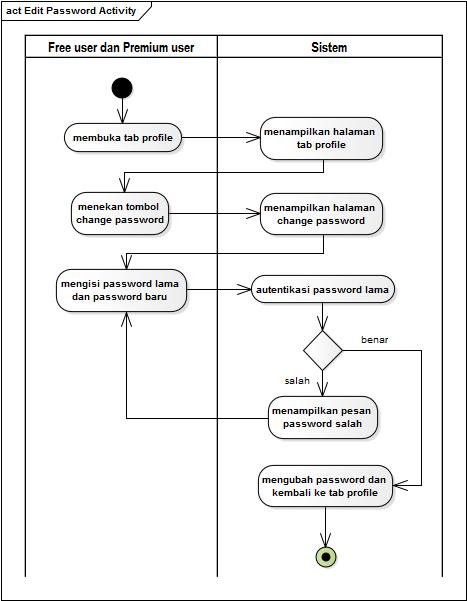
Pada gambar 3.8 menunjukkan diagram aktivitas dari *edit profile* pada aplikasiAndroid. Urutan aktivitas dimulai dari membuka halaman utama pada aplikasi dan berakhir dengan perubahan data pengguna.



Gambar 3.8 Diagram aktivitas edit profile pada aplikasi Android

1. **Diagram Aktivitas *Change Password* (Android)**

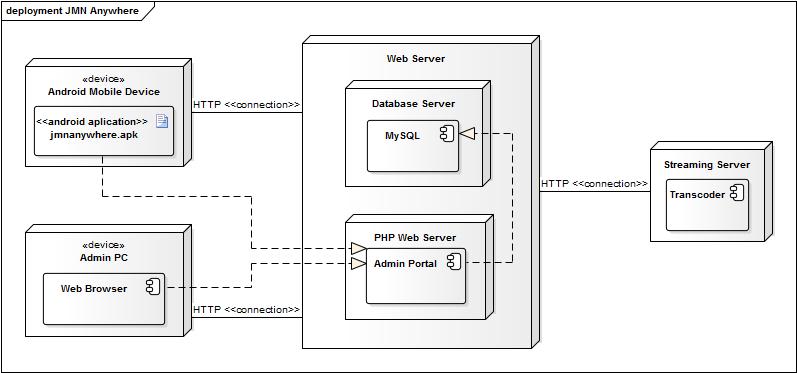
Pada gambar 3.9 menunjukkan diagram aktivitas dari *change password* pada aplikasiAndroid. Urutan aktivitas dimulai dari membuka halaman utama pada aplikasi dan berakhir dengan perubahan *password*.



Gambar 3.9 Diagram aktivitas change password pada aplikasi Android

#### *Deployment Diagram*

*Deployment diagram* menunjukkan bagaimana sistem yang akan digunakan secara fisik dalam lingkungan *hardware*. Tujuannya adalah untuk menunjukkan di mana berbagai komponen dari sistem fisik yang akan berjalan dan bagaimana mereka berkomunikasi satu sama lain.[9] Pada gambar 3.10 menunjukkan *deployment diagram* dari aplikasi JMN Anywhere.



Gambar 3.10 Deployment Diagram aplikasi JMN Anywhere

### Perancangan Basis Data

Dalam membuat aplikasi JMN Anywhere ini dibutuhkan suatu basis data untuk menyimpan data. Oleh karena itu, perancangan basis data sangat diperlukan. Perancangan basis data dimaksudkan untuk membuat basis data yang sesuai dengan kebutuhan sistem atau aplikasi yang akan dibuat. Sehingga, basis data yang ideal sangat dibutuhkan agar basis data yang dibuat lebih efisien dan sesuai dengan kebutuhan sistem. Proses perancangan basis data untuk aplikasi JMN Anywhere ini menggunakan metode pemodelan *Entity Relationship Diagram* (ERD). Adapun proses yang dilakukan dalam perancangan basis data dari aplikasi JMN Anywhere adalah sebagai berikut.

#### Menentukan Entitas

Pada tabel 3.21 menunjukkan deskripsi dari entitas pada basis data yang akan digunakan dalam aplikasi JMN Anywhere.

Tabel 3.21 Deskripsi entitas pada basis data

|  |  |  |
| --- | --- | --- |
| **No.** | **Entitas** | **Keterangan** |
| 1. | *genre* | Entitas yang berisi data *genre* |
| 2. | *genre\_movies* | Entitas yang berisi data *genre* yang dimiliki oleh setiap *movies* |
| 3. | iklan | Entitas yang berisi data iklan yang ditampilkan pada aplikasi JMN Anywhere pada Android |
| 4. | kategori\_*movie* | Entitas yang berisi data kategori *movies* |
| 5. | kategori\_tv | Entitas yang berisi data kategori televisi |
| 6. | *level* | Entitas yang berisi data tingkatan atau *level* pengguna yang membedakan *privileges* dalam mengakses aplikasi |
| 7. | *live\_tv* | Entitas yang berisi data *live tv* yang ditampilkan pada aplikasi |
| 8. | *movies* | Entitas yang berisi data *movies* yang ditampilkan pada aplikasi |
| 9. | *notif* | Entitas yang berisi data notifikasi yang telah dikirim ke pengguna |
| 10. | tbl\_kategori\_*movies* | Entitas yang berisi data kategori yang dimiliki oleh setiap *movies* |
| 11. | *users* | Entitas yang berisi data pengguna yang dapat mengakses aplikasi JMN Anywhere pada *web* maupun Android |

#### Menentukan Atribut dari Entitas

Setelah menentukan entitas, tahap selanjutnya adalah menentukan atribut dari entitas-entitas yang telah ditentukan yang disertai dengan jenis *key.*

1. **Entitas *Genre***

Pada tabel 3.22 menunjukkan rincian lengkap atribut dari entitas *genre*.

Tabel 3.22 Rincian lengkap atribut dari entitas genre

|  |  |  |  |
| --- | --- | --- | --- |
| **Atribut** | **Tipe Data** | **Key** | **Keterangan** |
| id\_*genre* | int(5) | *Primary key* | Id yang dimiliki *genre* |
| nama\_*genre* | varchar(20) |  | Nama *genre* yang digunakan |

1. **Entitas *Genre\_Movies***

Pada tabel 3.23 menunjukkan rincian lengkap atribut dari entitas *genre\_movies.*

Tabel 3.23 Rincian lengkap atribut dari entitas genre\_movies

|  |  |  |  |
| --- | --- | --- | --- |
| **Atribut** | **Tipe Data** | **Key** | **Keterangan** |
| id\_*genre\_movies* | int(5) | *Primary key* | Id yang dimiliki *genre movies* |
| id\_*genre* | varchar(5) | *Foreign key* | Id yang dimiliki oleh *genre* |
| id\_*movies* | varchar(25) | *Foreign key* | Id yang dimiliki oleh *movies* |

1. **Entitas Iklan**

Pada tabel 3.24 menunjukkan rincian lengkap atribut dari entitas iklan.

Tabel 3.24 Rincian lengkap atribut dari entitas iklan

|  |  |  |  |
| --- | --- | --- | --- |
| **Atribut** | **Tipe Data** | **Key** | **Keterangan** |
| id\_iklan | int(11) | *Primary key* | Id yang dimiliki *genre movies* |
| judul\_iklan | varchar(50) |  | Judul dari iklan |
| poster\_iklan | varchar(50) |  | Nama file poster yang disimpan |
| created\_at | datetime |  | Tanggal menambahkan data iklan |

1. **Entitas Kategori\_*Movie***

Pada tabel 3.25 menunjukkan rincian lengkap atribut dari entitas kategori\_*movie.*

Tabel 3.25 Rincian lengkap atribut dari entitas kategori\_movie

|  |  |  |  |
| --- | --- | --- | --- |
| **Atribut** | **Tipe Data** | **Key** | **Keterangan** |
| kode\_kategori\_*movie* | varchar(10) | *Primary key* | Kode yang dimiliki kategori *movie* |
| nama\_kategori\_*movie* | varchar(25) |  | Nama dari kategori *movie* |

1. **Entitas Kategori\_Tv**

Pada tabel 3.26 menunjukkan rincian lengkap atribut dari entitas kategori\_*tv.*

Tabel 3.26 Rincian lengkap atribut dari entitas kategori\_tv

|  |  |  |  |
| --- | --- | --- | --- |
| **Atribut** | **Tipe Data** | **Key** | **Keterangan** |
| kode\_kategori\_*tv* | varchar(10) | *Primary key* | Kode yang dimiliki kategori televisi |
| nama\_kategori\_*tv* | varchar(25) |  | Nama dari kategori televisi |

1. **Entitas *Level***

Pada tabel 3.27 menunjukkan rincian lengkap atribut dari entitas *level.*

Tabel 3.27 Rincian lengkap atribut dari entitas level

|  |  |  |  |
| --- | --- | --- | --- |
| **Atribut** | **Tipe Data** | **Key** | **Keterangan** |
| id\_*level* | varchar(5) | *Primary key* | Id yang dimiliki *level* |
| nama\_*level* | varchar(20) |  | Nama dari *level* yang digunakan |

1. **Entitas *Live\_Tv***

Pada tabel 3.28 menunjukkan rincian lengkap atribut dari entitas *live\_tv.*

Tabel 3.28 Rincian lengkap atribut dari entitas live\_tv

|  |  |  |  |
| --- | --- | --- | --- |
| **Atribut** | **Tipe Data** | **Key** | **Keterangan** |
| id\_*live\_tv* | int(11) | *Primary key* | Id yang dimiliki *live tv* |
| namatv | varchar(20) |  | Nama televisiyang disimpan |
| tv\_url | varchar(100) |  | Alamat *url* untuk mengakses televisi |
| *created\_at* | datetime |  | Tanggal menambahkan data televisi |
| *updated\_at* | datetime |  | Tanggal perubahan data televisi |
| logo | varchar(100) |  | Nama file logo televisi yang disimpan |
| kode\_kategori\_tv | varchar(25) | *Foreign key* | Kode kategori televisi |

1. **Entitas *Movies***

Pada tabel 3.29 menunjukkan rincian lengkap atribut dari entitas *movies.*

Tabel 3.29 Rincian lengkap atribut dari entitas movies

|  |  |  |  |
| --- | --- | --- | --- |
| **Atribut** | **Tipe Data** | **Key** | **Keterangan** |
| id\_*movie* | varchar(25) | *Primary key* | Id yang dimiliki *movies* |
| rating | varchar(10) |  | Rating yang dimiliki oleh *movies* |
| judul | varchar(50) |  | Nama judul yang dimiliki oleh *movies* |
| sinopsis | text |  | Sinopsis tentang *movies* |
| pemain | text |  | Daftar pemain yang ada dalam *movies* |
| sutradara | varchar(30) |  | Nama sutradara dari *movies* |
| penulis | varchar(50) |  | Nama penulis *movies* |
| durasi | text |  | Lama durasi *movies* |
| rilis | date |  | Tanggal rilis *movies* |
| poster | varchar(50) |  | Nama file poster yang disimpan |
| *trailer* | varchar(100) |  | Nama file *trailer* yang disimpan |
| *movie\_url* | varchar(100) |  | Nama file *movies* yang disimpan |
| *created\_at* | datetime |  | Tanggal menambahkan data televisi |
| *updated\_at* | datetime |  | Tanggal perubahan data televisi |

1. **Entitas Notif**

Pada tabel 3.30 menunjukkan rincian lengkap atribut dari entitas notif*.*

Tabel 3.30 Rincian lengkap atribut dari entitas notif

|  |  |  |  |
| --- | --- | --- | --- |
| **Atribut** | **Tipe Data** | **Key** | **Keterangan** |
| id\_notif | int(11) | *Primary key* | Id yang dimiliki notif |
| judul | varchar(30) |  | Judul notifikasi |
| pesan | text |  | Isi pesan dari notifikasi yang dikirim |
| file\_gambar | varchar(100) |  | Nama file gambar notifikasi yang disimpan |
| tgl\_kirim | datetime |  | Tanggal pengiriman notifikasi |

1. **Entitas Tbl\_Kategori\_*Movie***

Pada tabel 3.31 menunjukkan rincian lengkap atribut dari entitas tbl\_kategori\_*movie.*

Tabel 3.31 Rincian lengkap atribut dari entitas tbl\_kategori\_movie

|  |  |  |  |
| --- | --- | --- | --- |
| **Atribut** | **Tipe Data** | **Key** | **Keterangan** |
| id\_kategori | int(11) | *Primary key* | Id yang dimiliki tbl\_kategori*\_movie* |
| kode\_kategori\_*movie* | varchar(10) | *Foreign key* | Kode yang dimiliki oleh kategori *movie* |
| id\_*movie* | varchar(25) | *Foreign key* | Id yang dimiliki oleh *movies* |

1. **Entitas *Users***

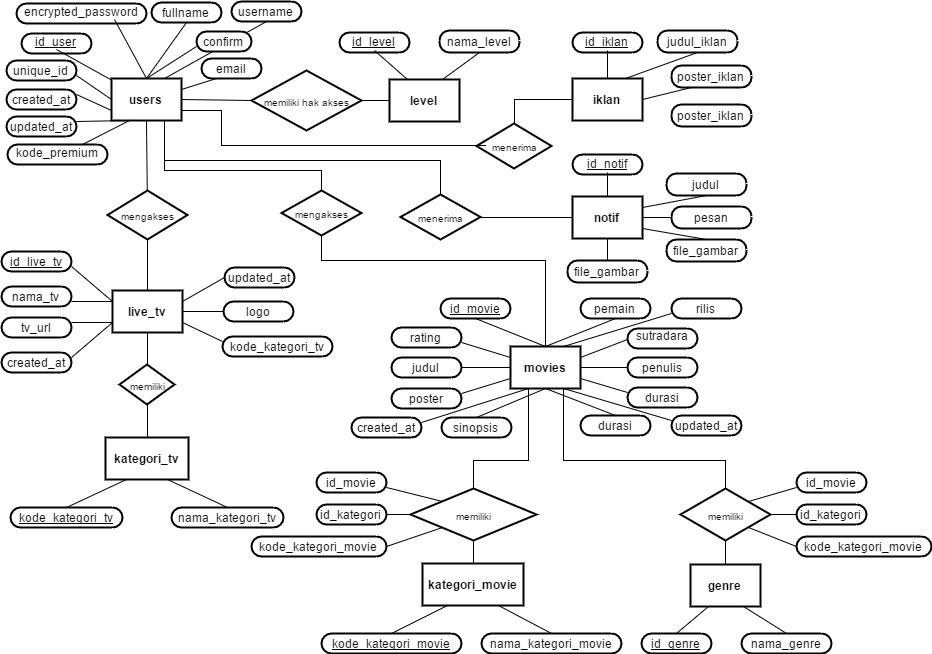
Pada tabel 3.32 menunjukkan rincian lengkap atribut dari entitas *users.*

Tabel 3.32 Rincian lengkap atribut dari entitas users

|  |  |  |  |
| --- | --- | --- | --- |
| **Atribut** | **Tipe Data** | **Key** | **Keterangan** |
| id\_*user* | int(11) | *Primary key* | Id yang dimiliki *user* |
| id\_*level* | int(5) | *Foreign key* | Id yang dimiliki oleh *level* |
| *unique\_id* | varchar(25) |  | Id unik yang dimiliki oleh *user* |
| *fullname* | varchar(100) |  | Nama lengkap yang dimiliki oleh *user* |
| *username* | varchar(100) |  | *Username* yang dimiliki oleh *user* |
| *email* | varchar(100) |  | Alamat *email* yang dimiliki oleh *user* |
| kode\_*premium* | binary(6) |  | Kode *premium* yang dimiliki oleh *user* |
| *encrypted\_password* | varchar(255) |  | *Hash password* yang dimiliki oleh *user* |
| *created\_at* | datetime |  | Tanggal menambahkan data *user* |
| *updated\_at* | datetime |  | Tanggal perubahan data *user* |
| *confirm* | varchar(5) |  | Status konfirmasi alamat *email user* |

#### Menentukan Relasi antar Entitas

Setelah menentukan atribut pada masing-masing entitas, tahap selanjutnya adalah menetukan relasi yang terjadi antar entitas-entitas tersebut. Pada gambar 3.11 menunjukkan relasi antar entitas secara keseluruhan.



Gambar 3.11 Relasi antar entitas secara keseluruhan

ERD dari sistem memiliki sebelas entitas yang terlibat yaitu *genre*, *genre\_movies*, iklan, kategori\_*movie*, kategori\_tv, *level*, *live\_tv*, *movies*, notif, tbl\_kategori\_*movie* dan *user*. Masing-masing entitas memiliki atribut yang berbeda-beda. Relasi antara *user* dan *level* termasuk dalam jenis *many to one,* karena setiap *user* hanya dapat memiliki satu *level*. Relasi antara *user* dan iklan termasuk dalam jenis *one to many*, karena setiap *user* dapat menerima beberapa iklan. Relasi antara *user* dan notif termasuk dalam jenis *one to one*, karena setiap *user* hanya dapat menerima satu notif. Relasi antara *live\_tv* dan kategori\_tv termasuk dalam jenis *many to one,* karena setiap *live\_tv* hanya dapat memiliki satu kategori\_tv. Relasi antara *movies* dan *genre* termasuk dalam jenis *many to many*, karena setiap *movies* memiliki banyak *genre* dan sebaliknya. Relasi antara *movies* dan kategori*\_movie* termasuk dalam jenis *many to many,* karena setiap *movies* memiliki banyak kategori\_*movie* dan sebaliknya.

# BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

## Implementasi

Implementasi merupakan suatu tahap dimana penerapan dari proses perancangan yang telah dibuat sebelumnya pada aplikasi. Pada tahap ini dilakukan implementasi terhadap aplikasi JMN Anywhere. Implementasi ini bertujuan agar terwujud aplikasi yang memiliki fungsi dan antarmuka sesuai dengan rancangan desain sistem. Implementasi dilakukan pada setiap kegiatan yang dikerjakan masing-masing pengguna pada *website* maupun *mobile.*

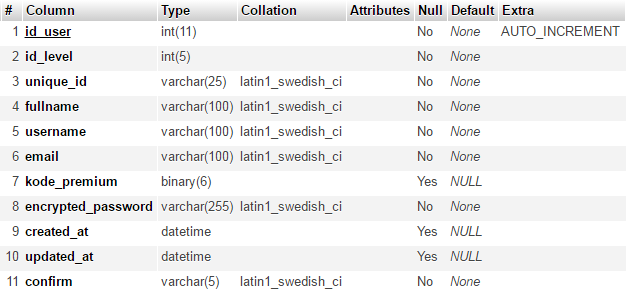
### Implementasi Basis Data

Implementasi basis data dilakukan sesuai dengan tahap perancangan sebelumnya dengan melakukan proses pembuatan tabel-tabel pada basis data menggunakan basis data MySQL.

1. **Tabel *Users***

Pada tabel 4.1 menunjukkan struktur dari tabel *users* yang terdiri dari beberapa kolom yaitu *id\_user,* *id\_level, unique\_id, fullname, username, email,* kode\_*premium*, *encrypted\_password,* *created\_at, updated\_at, confirm*.

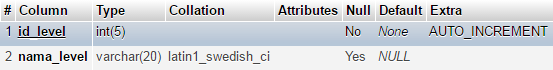
Tabel 4.1 Struktur tabel users



1. **Tabel *Level***

Pada tabel 4.2 menunjukkan struktur tabel *level* yang terdiri dari dua kolom yaitu *id\_level* dan nama\_*level*.

Tabel 4.2 Struktur tabel level



1. **Tabel Iklan**

Pada tabel 4.3 menunjukkan struktur dari tabel iklan yang terdiri dari beberapa kolom yaitu id\_iklan, judul\_iklan, poster\_iklan, *created\_at.*

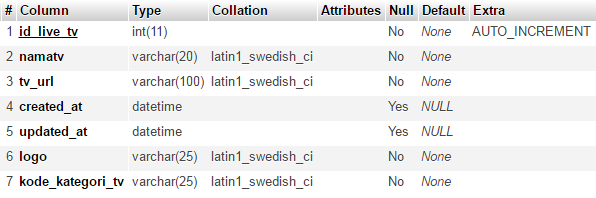
Tabel 4.3 Struktur tabel iklan



1. **Tabel *Live\_TV***

Pada tabel 4.4 menunjukkan struktur dari tabel *live tv* yang terdiri dari beberapa kolom yaitu id\_*live\_tv*, namatv, *tv*\_*url,* *created\_at, updated\_at,* logo*,* kode*\_*kategori*\_tv.*

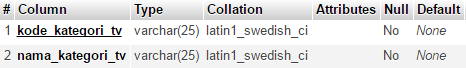
Tabel 4.4 Struktur tabel live\_tv



1. **Tabel Kategori\_TV**

Pada tabel 4.5 menunjukkan struktur dari tabel kategori\_*tv* yang terdiri dari dua kolom yaitu kode\_kategori\_*tv* dan nama\_kategori\_*tv.*

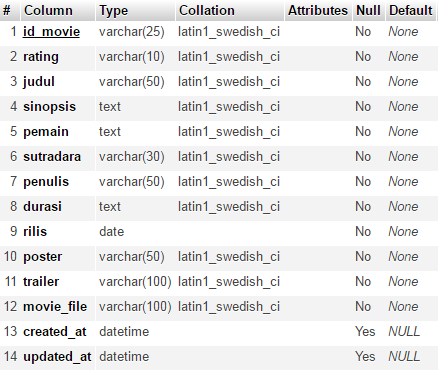
Tabel 4.5 Struktur tabel kategori\_tv



1. **Tabel Movies**

Pada tabel 4.6 menunjukkan struktur dari tabel movies yang terdiri dari beberapa kolom yaitu id\_movie, rating, judul, sinopsis, pemain, sutradara, penulis, durasi, rilis, poster, trailer, movie\_url, created\_at, updated\_at.

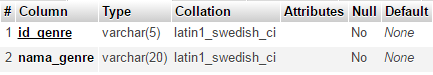
Tabel 4.6 Struktur tabel movies



1. **Tabel Genre**

Pada tabel 4.7 menunjukkan struktur dari tabel genre yang terdiri dari dua kolom yaitu id\_genre dan nama\_genre.

Tabel 4.7 Struktur tabel genre



1. **Tabel Genre\_Movies**

Pada tabel 4.8 menunjukkan struktur dari tabel *genre\_movies* yang terdiri dari tiga yaitu *id\_genre\_movies, id\_genre, dan id\_movie.*

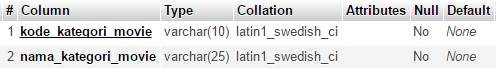
Tabel 4.8 Struktur tabel genre\_movies



1. **Tabel Kategori\_Movie**

Pada tabel 4.9 menunjukkan struktur dari tabel kategori­\_movie yang terdiri dari beberapa kolom yaitu kode\_kategori\_movie, nama\_kategori\_movie.

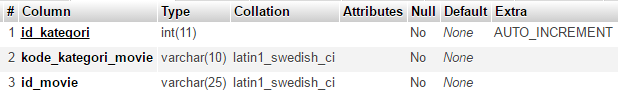
Tabel 4.9 Struktur tabel kategori\_movie



1. **Tabel Tbl\_Kategori\_Movie**

Pada tabel 4.10 menunjukkan struktur dari tabel tbl\_kategori\_*movie* yang terdiri dari tiga kolom yaitu ­­*id\_kategori,* kode\_kategori\_*movie*, dan *id\_movie.*

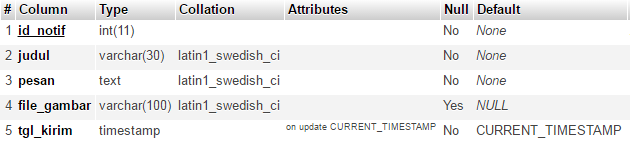
Tabel 4.10 Struktur tabel tbl\_kategori\_movie



1. **Tabel Notif**

Pada tabel 4.11 menunjukkan struktur dari tabel notif yang terdiri dari beberapa kolom yaitu id\_notif, judul, pesan, file\_gambar, tgl\_kirim.

Tabel 4.11 Struktur tabel notif



### Implementasi Antarmuka Aplikasi JMN Anywhere pada Android

Pada bagian ini menjelaskan tentang tampilan antarmuka aplikasi pada Android yang telah dibangun. Tampilan antarmuka pada aplikasi ini dibuat menggunakan kode program sesuai dengan perancangan desain antarmuka aplikasi yang telah dibangun sehingga menghasilkan antarmuka yang diharapkan. Aplikasi pada Android ini penggunanya yaitu *free user* dan *premium user.*

1. **Halaman *Splash Screen***

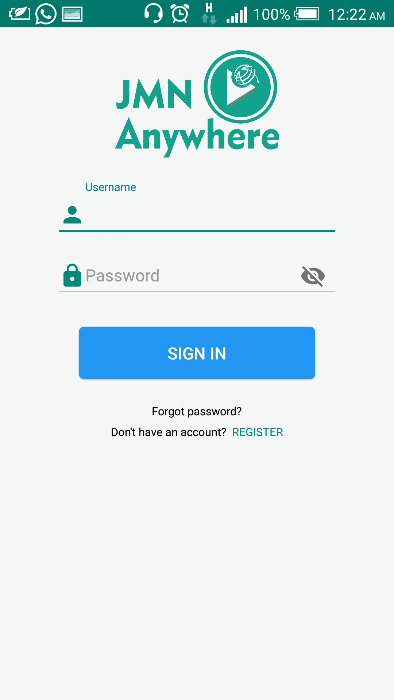
*Splash screen* merupakan tampilan awal sebelum masuk ke halaman *login* atau ke halaman utama dari aplikasi. Tampilan *splash screen* ini muncul ketika pertama kali membuka aplikasi dan memiliki durasi penampilan selama 4 detik. Pada halaman *splash screen* menampilkan logo dan nama aplikasi. Pada gambar 4.1 menunjukkan tampilan dari halaman *splash screen* aplikasi JMN Anywhere pada Android*.*



Gambar 4.1 Tampilan halaman splash screen aplikasi JMN Anywhere pada Android

1. **Halaman *Login***

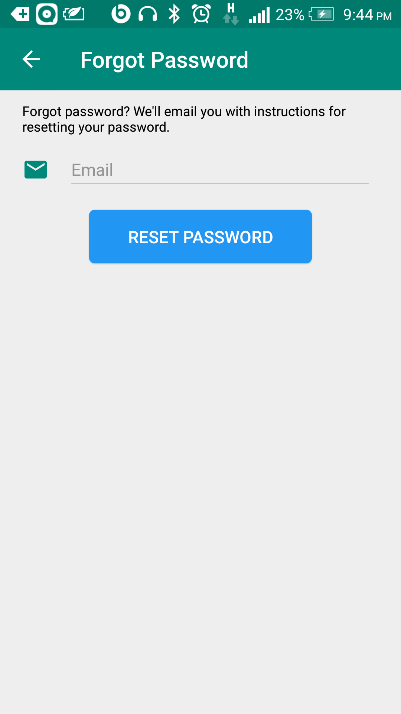
Halaman *login* ini merupakan halaman yang digunakan pengguna untuk masuk ke halaman utama sistem dari aplikasi JMN Anywhere. Pada halaman *login* ini terdapat dua *input form* yang memiliki fungsi berbeda yaitu *form username* berfungsi untuk menampung *username* dari pengguna dan *form password* berfungsi untuk menampung *password* dari pengguna. Selain itu, terdapat dua *textview* yang dapat diklik yaitu *“Forgot password?”* yang ketika diklik akan menuju halaman *forgot password* dan “*Don’t have an account? REGISTER”* yang ketika diklik menuju halaman *register*. Terdapat satu buah tombol *“Login”* yang digunakan untuk memulai autentikasi *username* dan *password* dan jika benar akan membawa pengguna ke halaman utama aplikasi sesuai status penggunanya. Pada gambar 4.2 menunjukkan tampilan dari halaman *login* aplikasi JMN Anywhere pada Android*.*



Gambar 4.2 Tampilan halaman login aplikasi JMN Anywhere pada Android

1. **Halaman *Forgot Password***

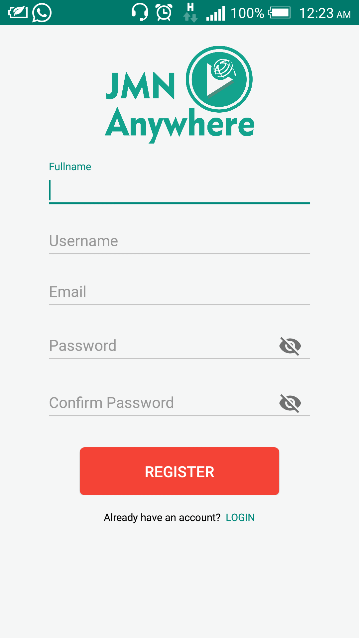
Halaman *forgot password* digunakan pengguna jika pengguna mengalami lupa *password* ketika akan melakukan *login.* Pada halaman *forgot password* terdapat satu *input form* yang memiliki fungsi untuk menampung alamat *email* dari pengguna dan tombol *“Reset Password”* yang digunakan untuk mengirim *link reset password* ke alamat *email* pengguna. Pada gambar 4.3 menunjukkan tampilan dari halaman *forgot password* aplikasi JMN Anywhere pada Android.



Gambar 4.3 Tampilan halaman forgot password aplikasi JMN Anywhere pada Android

1. **Halaman *Register***

Halaman *register* merupakan halaman yang digunakan pengguna baru untuk mendaftar sebagai *free user* agar dapat menggunakan apilkasi JMN Anywhere. Pada halaman *register* terdapat lima *input form* yang terdiri dari *fullname* yang digunakan untuk menampung nama lengkap pengguna, *username* yang digunakan untuk menampung *username* pengguna untuk melakukan *login,* *email* yang digunakan untuk menampung alamat *email* pengguna, *password* digunakan untuk menampung *password* pengguna untuk melakukan *login,* dan *confirm password* untuk menampung pengulangan *password* yang pengguna masukan. Selain itu, terdapat sebuah tombol *“Register”* dan sebuah *textview* yang dapat diklik. Tombol *“Register”* digunakan untuk mulai melakukan registrasi dan menyimpan data pada *database* sedangkan *textview* *“Already have an account? Login”* digunakan untuk menuju ke halaman *login* jika sudah memiliki akun. Pada gambar 4.4 menunjukkan tampilan dari halaman *register* aplikasi JMN Anywhere pada Android.



Gambar 4.4 Tampilan halaman register aplikasi JMN Anywhere pada Android

1. **Halaman *Home Free User***

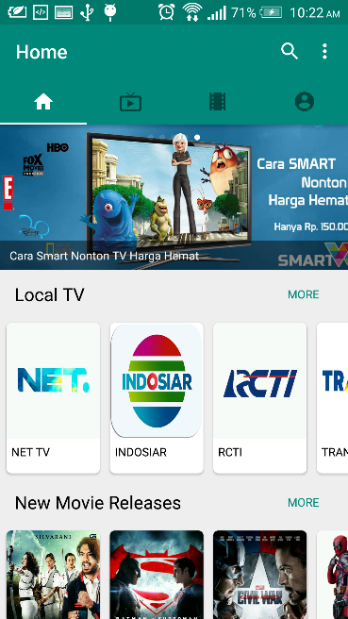
Halaman *home free user* merupakan tampilan halaman *home* untuk pengguna dengan status *free user.* Pada halaman *home free user* terdapat *cardview list* yang berfungsi hanya untuk menampilkan daftar *channel* televisi dengan kategori lokal dan *image slider* untuk menampilkan iklan. Pada gambar 4.5 menunjukkan tampilan dari halaman *home free user* aplikasi JMN Anywhere pada Android.



Gambar 4.5 Tampilan halaman home free user aplikasi JMN Anywhere pada Android

1. **Halaman *Home Premium User***

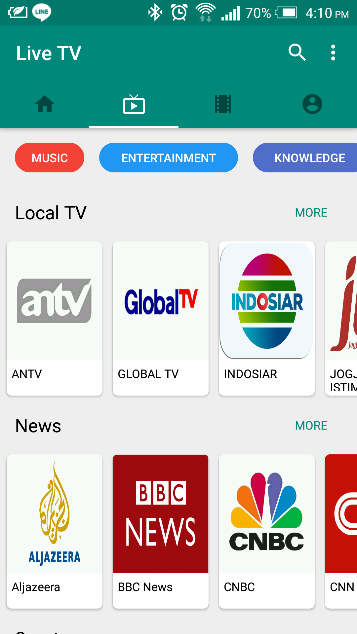
Halaman *home premium user* merupakan tampilan halaman *home* untuk pengguna dengan status *premium user.* Pada halaman *home premium user* terdapat beberapa *cardview* untuk menampilkan dua kategori *channel* televisi yaitu *local tv* dan *news* dan dua kategori *movies* yaitu *new movie releases* dan *indonesian*. Selain itu, terdapat *textview* *“More”* yang dapat diklik berfungsi menampilkan halaman *more* untuk melihat lebih banyak data dan terdapat *image slider* untuk menampilkan iklan. Pada gambar 4.6 menunjukkan tampilan dari halaman *home premium user* aplikasi JMN Anywhere pada Android*.*



Gambar 4.6 Tampilan halaman home premium user aplikasi JMN Anywhere pada Android

1. **Halaman *Live TV***

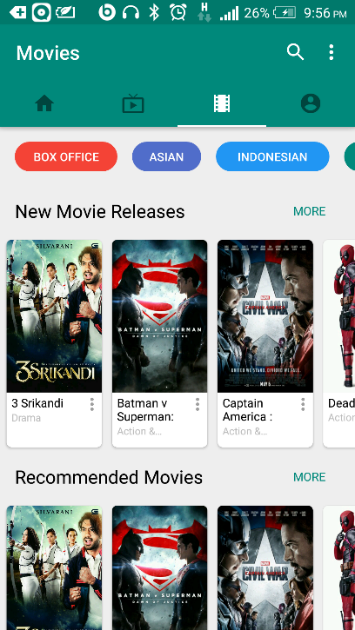
Halaman *live tv* merupakan halaman dari tab menu *live tv* yang digunakan untuk menampilkan daftar *live tv.* Pada halaman *live tv* terdapat *cardview* untuk menampilkan daftar *channel* televisi dengan kategori *local tv*, *news,* dan *sport*. Selain itu, terdapat *textview* *“More”* yang dapat diklik berfungsi menampilkan halaman *more* untuk melihat lebih banyak data dan terdapat daftar pilihan kategori *channel* televisi. Pada gambar 4.7 menunjukkan tampilan dari halaman *live tv* aplikasi JMN Anywhere pada Android.



Gambar 4.7 Tampilan halaman live tv aplikasi JMN Anywhere pada Android

1. **Halaman *Movies***

Halaman *movies* hanya ada pada akun *premium*. Pada halaman *movies* ini menampilkan daftar *movies*. Terdapat *cardview* untuk menampilkan daftar *movies* dengan kategori *new movie releases, recommended movies,* dan *animation*. Selain itu, terdapat *textview* *“More”* yang dapat diklik berfungsi menampilkan halaman *more* untuk melihat lebih banyak data dan terdapat daftar pilihan kategori *movies*. Pada gambar 4.8 menunjukkan tampilan dari halaman *movies* aplikasi JMN Anywhere pada Android.



Gambar 4.8 Tampilan halaman movies aplikasi JMN Anywhere pada Android

1. **Halaman *Movie Info***

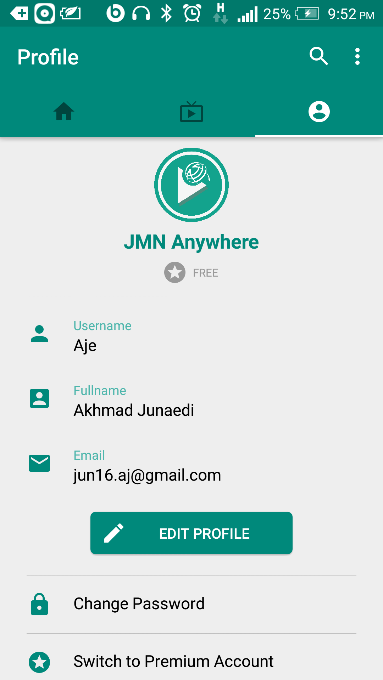
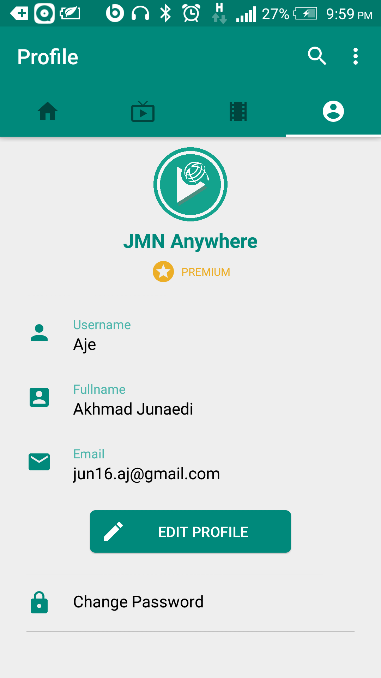
Pada halaman *movie info* menampilkan data lengkap dari *movie* sesuai dengan yang dipilih. Data *movie* yang ditampilkan tentang *genre,* durasi *movie, rating,* tanggal rilis, *trailer,* sinopsis, pemain, sutradara, dan penulis skenario. Pada gambar 4.9 menunjukkan tampilan dari halaman *movie info* aplikasi JMN Anywhere pada Android*.*



Gambar 4.9 Tampilan halaman movie info aplikasi JMN Anywhere pada Android

1. **Halaman *Profile***

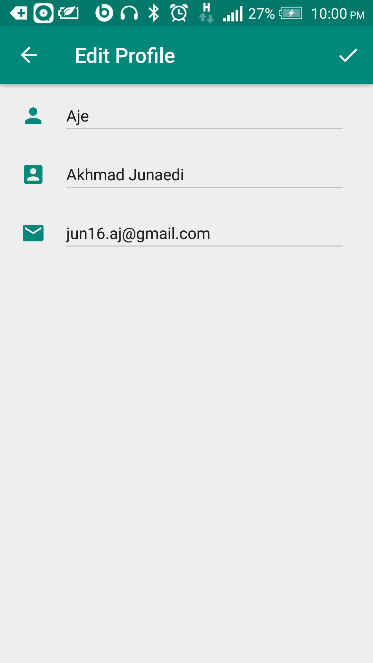
Pada halaman *profile* menampilkan data lengkap pengguna yaitu *username, fullname,* alamat *email,* dan status akun pengguna. Selain itu, terdapat dua *textview* yang dapat diklik yaitu *textview “Change Password”* untuk menuju ke halaman ganti *password* dan *textview* *“Switch to Premium Account”* yang hanya terdapat pada akun dengan status *free user* yang digunakan untuk menuju halaman *“Switch to Premium”*. Sebuah tombol *“Edit Profile”* digunakan untuk menuju halaman *edit profile.* Pada gambar 4.10 menunjukkan tampilan dari halaman *profile* aplikasi JMN Anywhere pada Android baik dari akun *free user* maupun *premium user*.

Gambar 4.10 Tampilan halaman profile free user (kiri) dan premium user (kanan) aplikasi JMN Anywhere pada Android

1. **Halaman *Edit Profile***

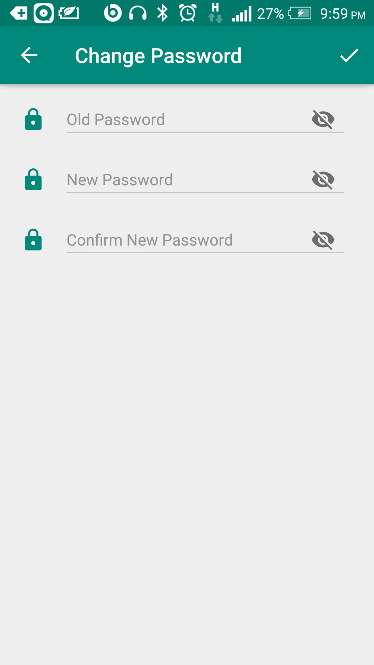
Pada halaman *edit profile* digunakan untuk mengubah data pengguna. Terdapat *input form* untuk menampung *username*, *fullname,* dan alamat *email* pengguna yang digunakan untuk mengubah data pengguna. Selain itu, terdapat tombol *checklist* untuk menyimpan data yang telah diubah ke *database.* Pada gambar 4.11 menunjukkan tampilan dari halaman *edit profile* aplikasi JMN Anywhere pada Android*.*



Gambar 4.11 Tampilan halaman edit profile aplikasi JMN Anywhere pada Android

1. **Halaman *Change Password***

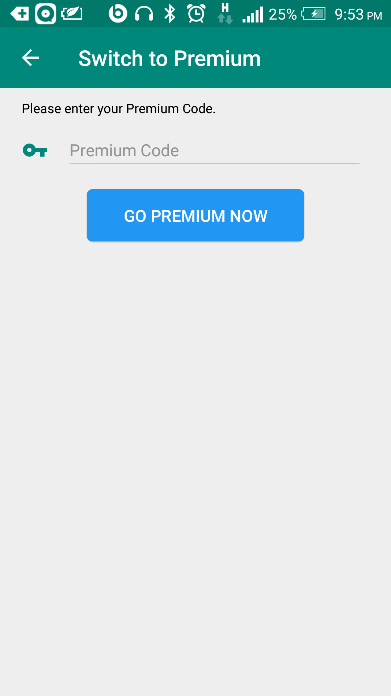
Pada halaman *change password* digunakan untuk mengubah kata sandi pengguna. Terdapat tiga *input form* untuk menampung kata sandi lama, kata sandi baru, dan konfirmasi kata sandi baru. Selain itu, terdapat tombol *checklist* untuk mengubah kata sandi lama dengan kata sandi baru dan menyimpan ke *database*. Pada gambar 4.12 menunjukkan tampilan dari halaman *change password* aplikasi JMN Anywhere pada Android.



Gambar 4.12 Tampilan halaman change password aplikasi JMN Anywhere pada Android

1. **Halaman *Switch to Premium***

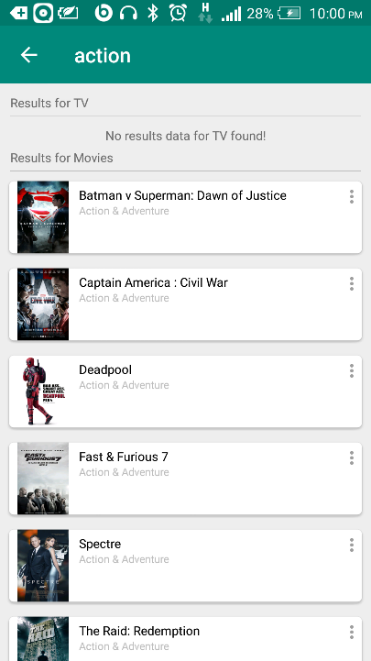
Pada halaman *switch to premium* digunakan untuk mengubah status akun pengguna dari *free user* ke *premium user* dengan memasukkan kode *premium*. Terdapat sebuah *input form* untuk menampung kode *premium* dan tombol *“Go Premium Now”* untuk melakukan validasi kode *premium*. Pada gambar 4.13 menunjukkan tampilan dari halaman *switch to premium* aplikasi JMN Anywhere pada Android.



Gambar 4.13 Tampilan halaman switch to premium aplikasi JMN Anywhere pada Android

1. **Halaman Hasil Pencarian**

Pada halaman hasil pencarian digunakan untuk menampilkan hasil pencarian data *live tv* maupun *movies* sesuai dengan kata kunci yang dimaksud. Hasil pencarian data ditampilkan pada *cardview list*. Jika data yang dicari tidak ditemukan akan muncul pesan. Pada gambar 4.14 menunjukkan tampilan dari halaman hasil pencarian aplikasi JMN Anywhere pada Android.



Gambar 4.14 Tampilan halaman hasil pencarian aplikasi JMN Anywhere pada Android

1. **Halaman *Video Player***

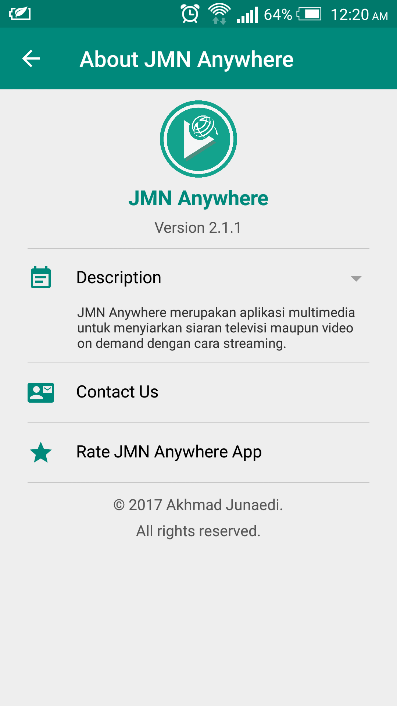
Pada halaman *video player* digunakan untuk menampilkan *channel live tv*  maupun *movies* yang dipilih untuk dimainkan. Terdapat tombol *play, pause, forward,* dan *backward*. *Video player* ini dapat memainkan *channel live tv* maupun *movie* dengan layar penuh. Pada gambar 4.15 menunjukkan tampilan dari halaman *video player* aplikasi JMN Anywhere pada Android.



Gambar 4.15 Tampilan halaman video player aplikasi JMN Anywhere pada Android

1. **Halaman *About* JMN Anywhere**

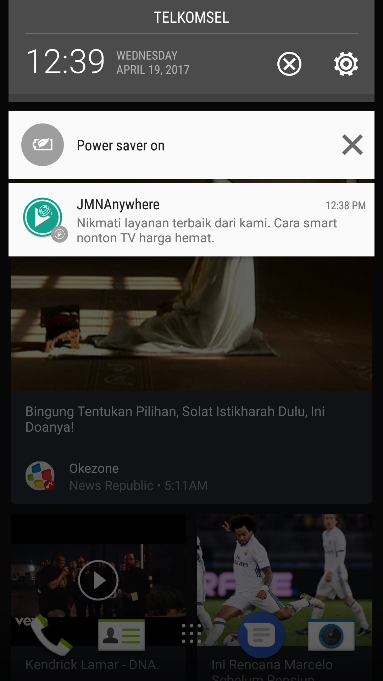
Pada halaman *about* JMN Anywhere digunakan untuk menampilkan deskripsi singkat mengenai aplikasi JMN Anywhere, kontak kami, dan beri kami nilai. Pada gambar 4.16 menunjukkan tampilan dari halaman *about* JMN Anywhere aplikasi JMN Anywhere pada Android*.*



Gambar 4.16 Tampilan halaman about JMN Anywhere pada Android

1. **Tampilan Notifikasi**

Notifikasi akan muncul pada perangkat Android ketika pengguna sudah melakukan *login*. Notifikasi yang diterima pengguna yaitu berupa informasi penting atau informasi berupa iklan. Pada gambar 4.17 menunjukkan tampilan dari notifikasi baik dengan gambar maupun tanpa gambar.

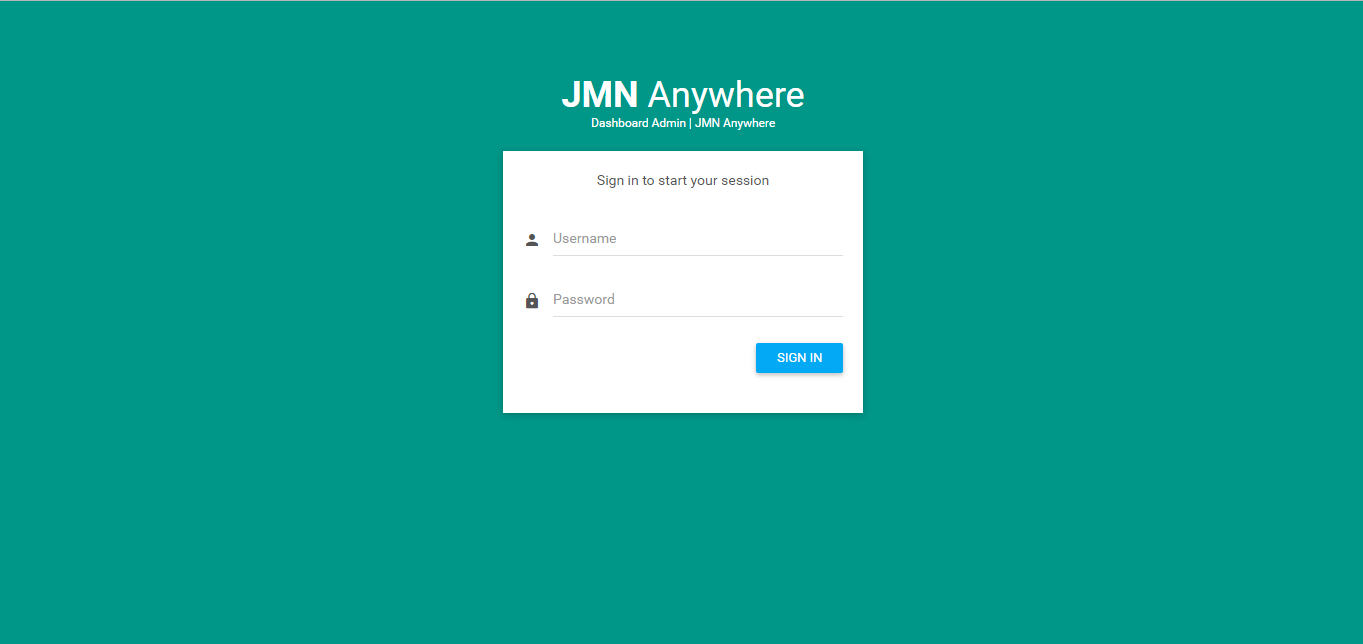
Gambar 4.17 Tampilan notifikasi tanpa gambar (kiri) dan dengan gambar (kanan) aplikasi JMN Anywhere pada Android

### Implementasi Antarmuka Aplikasi *Dashboard* Admin JMN Anywhere pada *Web*

Pada bagian ini menjelaskan tentang tampilan antarmuka aplikasi pada *web* yang telah dibangun. Tampilan antarmuka pada aplikasi ini dibuat menggunakan kode program sesuai dengan perancangan desain antarmuka aplikasi yang telah dibangun sehingga menghasilkan antarmuka yang diharapkan. Aplikasi pada *web* ini penggunanya hanya admin.

1. **Halaman *Login***

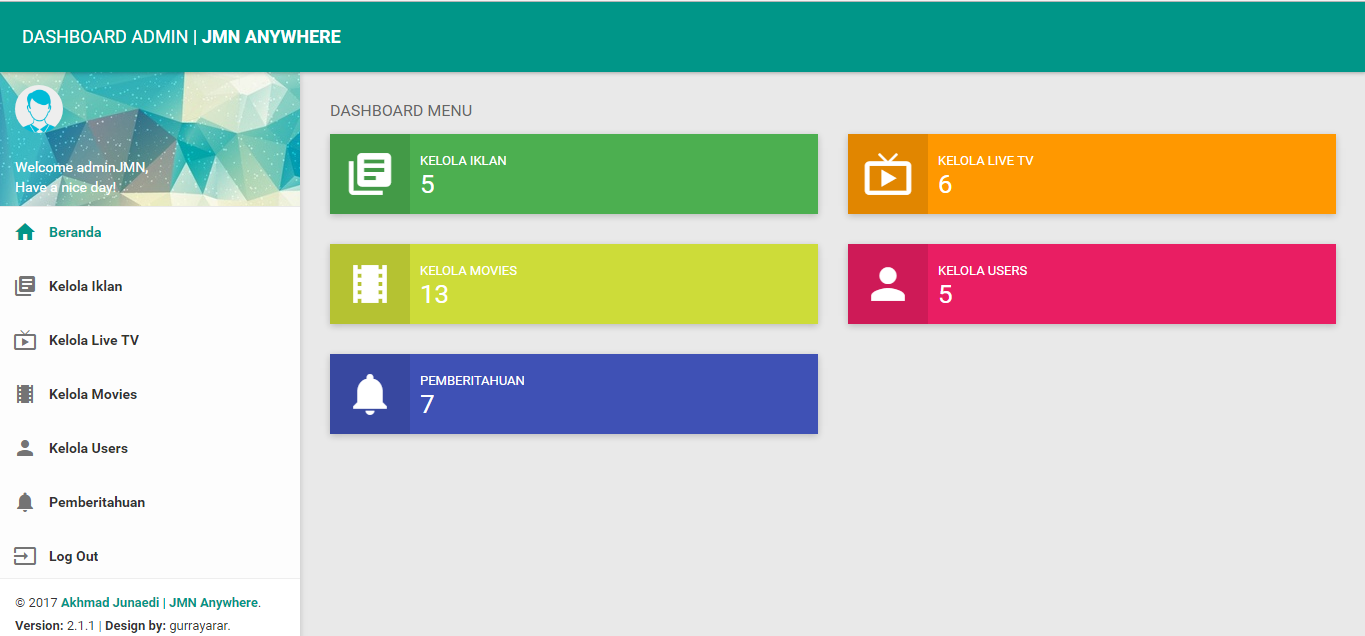
Halaman *login* merupakan halaman pertama yang tampil ketika awal membuka aplikasi JMN Anywhere melalui *web*. Pada halaman *login* menampilkan dua *input form* yang berfungsi untuk menampung *username* dan *password* pengguna yaitu admin. Dan terdapat sebuah tombol untuk memulai autentikasi *username* dan *password* pengguna untuk melanjutkan ke halaman utama sistem. Pada gambar 4.18 menunjukkan tampilan dari halaman *login* aplikasi JMN Anywhere pada *web*.



Gambar 4.18 Tampilan halaman login aplikasi JMN Anywhere pada web

1. **Halaman Utama**

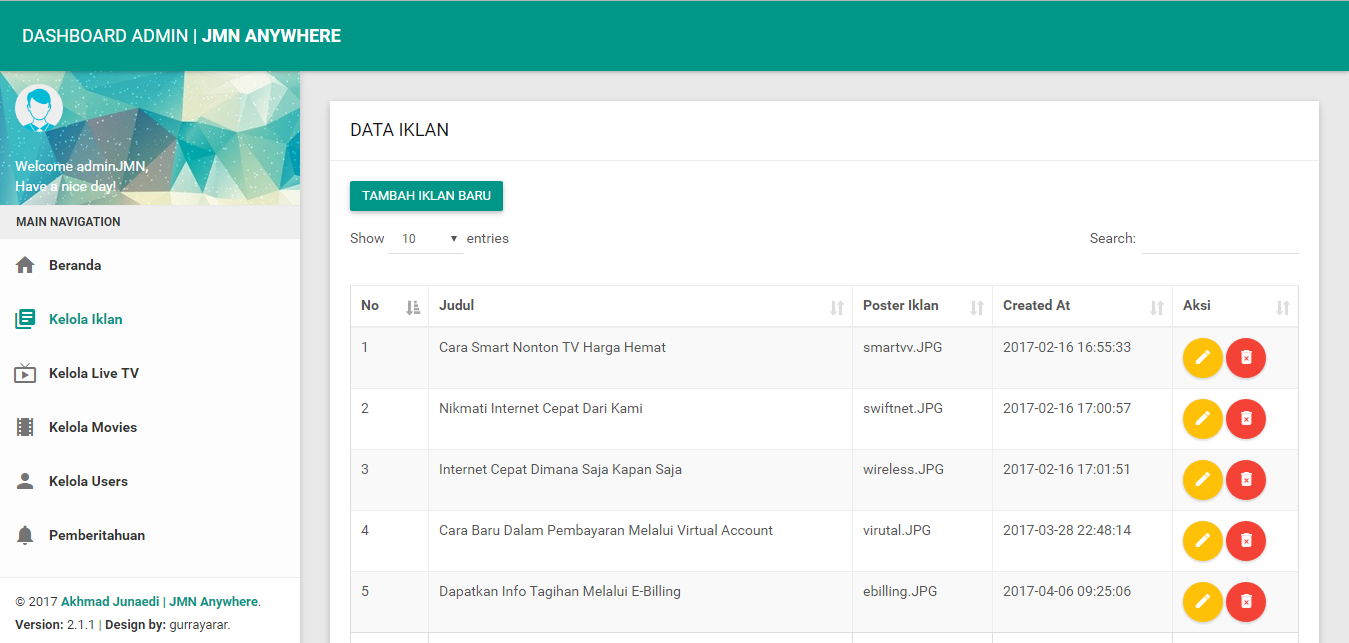
Pada halaman utama terdapat menu navigasi untuk mengelola berbagai data seperti data iklan, *live tv, movies, user,* pemberitahuan, dan tombol *logout*. Selain itu, pada halaman ini terdapat info *box* untuk menampilkan jumlah data yang tersimpan. Pada gambar 4.19 menunjukkan tampilan dari halaman utama aplikasi JMN Anywhere pada *web*.



Gambar 4.19 Tampilan halaman utama aplikasi JMN Anywhere pada web

1. **Halaman Kelola Iklan**

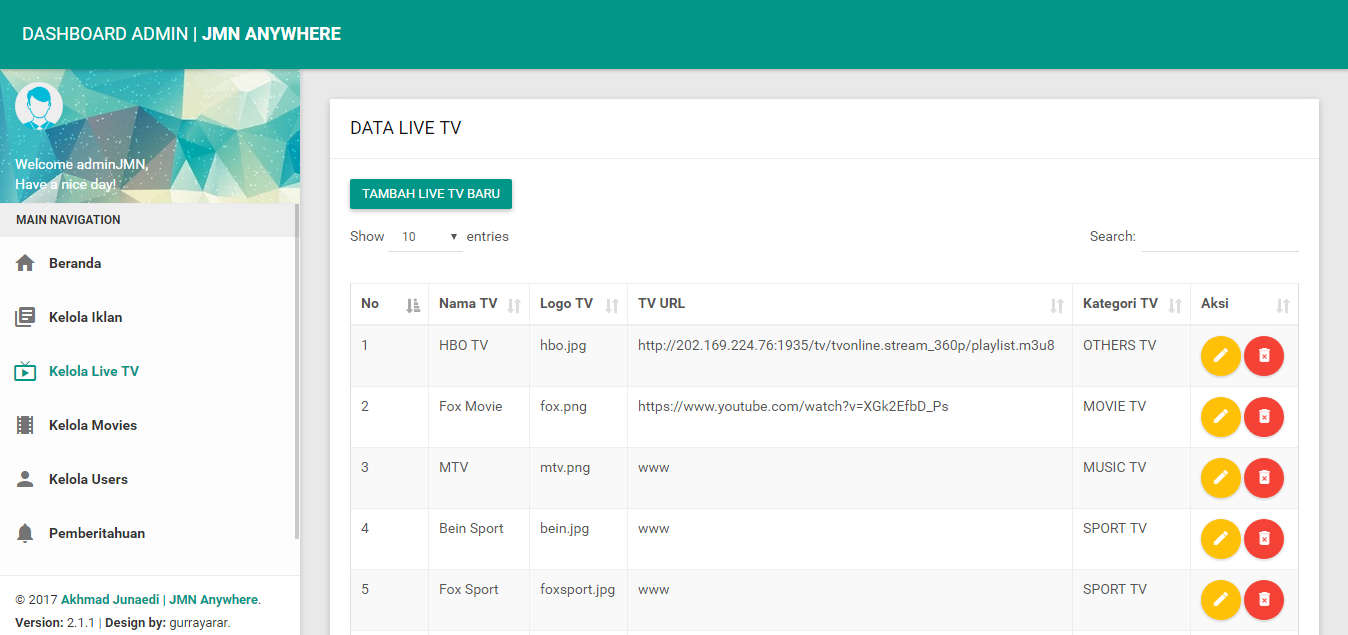
Pada halaman kelola iklan digunakan untuk menampilkan semua data iklan yang tersimpan dalam *database* dan mengelola data iklan seperti tambah data, *edit* data, dan hapus data. Terdapat beberapa tombol yang memiliki fungsi berbeda yaitu tombol tambah data iklan untuk menambah data iklan baru, tombol *edit* data untuk mengubah data pada halaman *edit* data, tombol hapus untuk menghapus data dari *database*. Pada gambar 4.20 menunjukkan tampilan dari halaman kelola iklan aplikasi JMN Anywhere pada *web*.



Gambar 4.20 Tampilan halaman kelola iklan aplikasi JMN Anywhere pada web

1. **Halaman Kelola *Live TV***

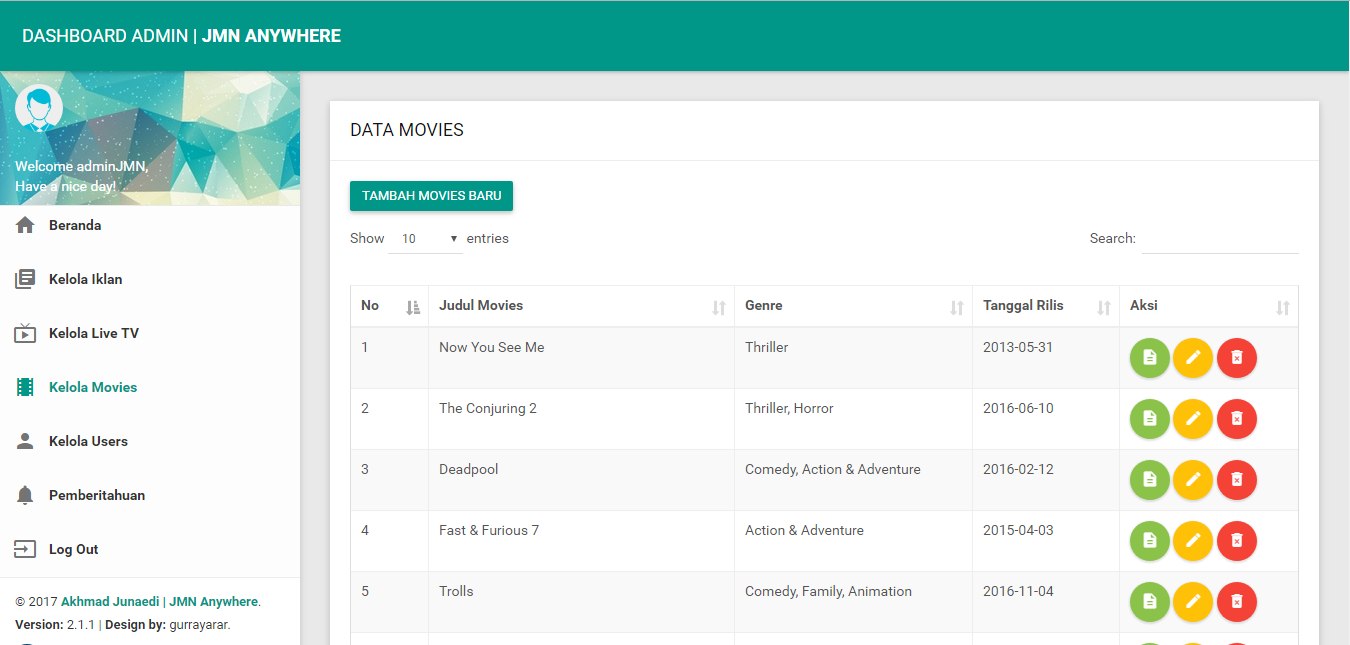
Pada halaman kelola *live tv* digunakan untuk menampilkan semua data *live tv* yang tersimpan dalam *database* dan mengelola data *live tv* seperti tambah data, *edit* data, dan hapus data. Terdapat beberapa tombol yang memiliki fungsi berbeda yaitu tombol tambah data *live tv* untuk menambah data *live tv* baru, tombol *edit* data untuk mengubah data pada halaman *edit* data, tombol hapus untuk menghapus data dari *database*. Pada gambar 4.21 menunjukkan tampilan dari halaman kelola *live tv* aplikasi JMN Anywhere pada *web.*



Gambar 4.21 Tampilan halaman kelola live tv aplikasi JMN Anywhere pada web

1. **Halaman Kelola *Movies***

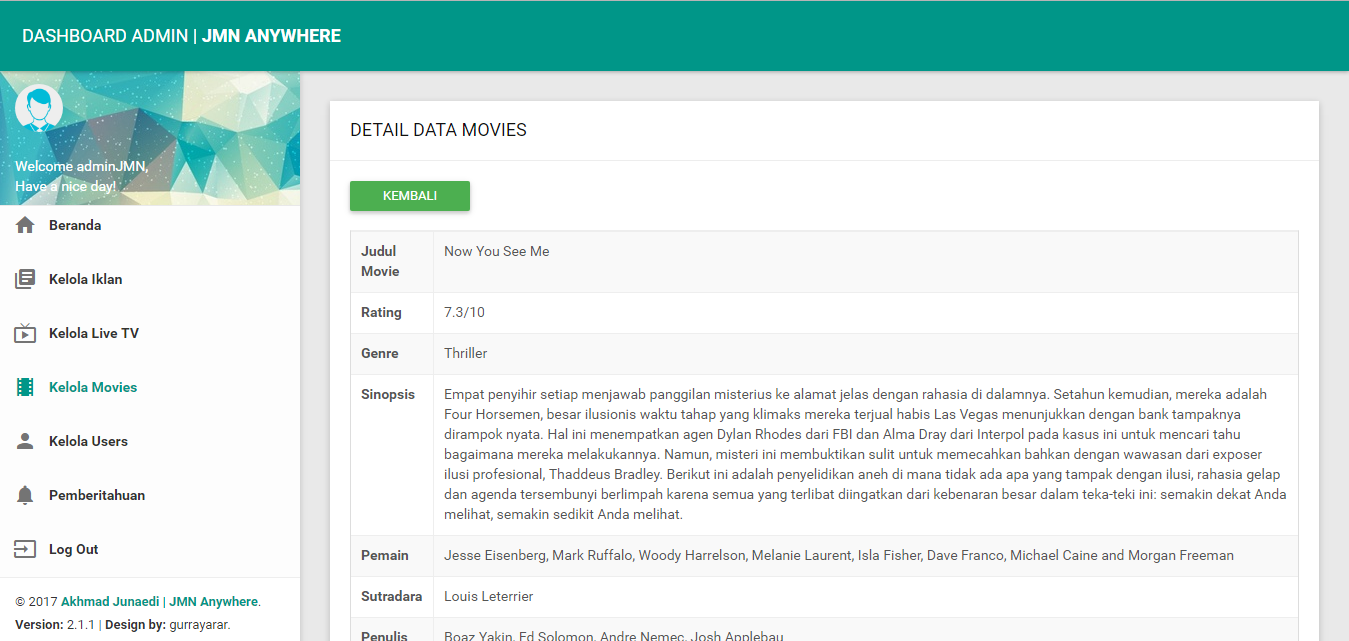
Pada halaman kelola *movies* digunakan untuk menampilkan semua data *movies* yang tersimpan dalam *database* hanya tentang judul, nama genre, dan tanggal rilis serta mengelola data *movies* seperti tambah data, *edit* data, dan hapus data. Terdapat beberapa tombol yang memiliki fungsi berbeda yaitu tombol tambah data *movies* untuk menambah data *movies* baru, tombol *edit* data untuk mengubah data pada halaman *edit* data, tombol hapus untuk menghapus data dari *database*. Pada gambar 4.22 menunjukkan tampilan dari halaman kelola *movies* aplikasi JMN Anywhere pada *web*.



Gambar 4.22 Tampilan halaman kelola movies aplikasi JMN Anywhere pada web

1. **Halaman *Detail Movies***

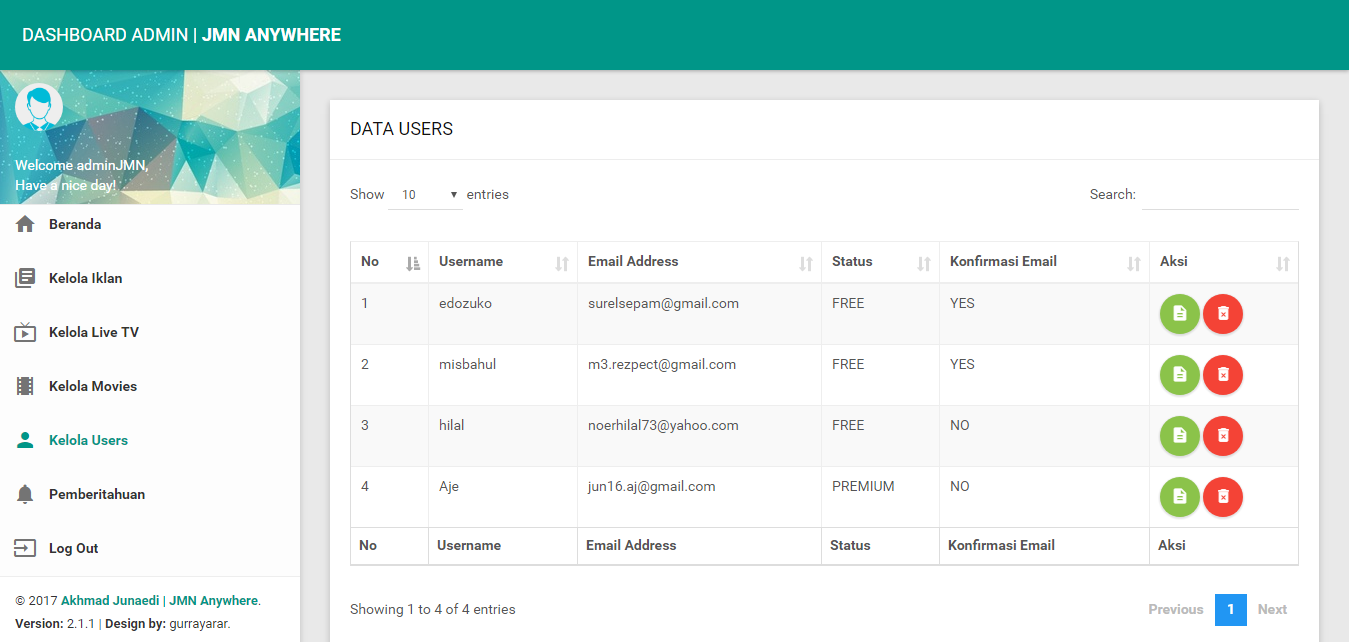
Pada halaman *detail movies* digunakan untuk menampilkan data *movies* secara lengkap. Data yang ditampilkan yaitu judul, rating, *genre,* sinopsis, pemain, sutradara, penulis, durasi, tanggal rilis, nama *file* poster, nama *file trailer,* nama *file movies* dan kategori *movies*. Pada gambar 4.23 menunjukkan tampilan dari halaman *detail movies* aplikasi JMN Anywhere pada *web*.



Gambar 4.23 Tampilan halaman detail movies aplikasi JMN Anywhere pada web

1. **Halaman Kelola *User***

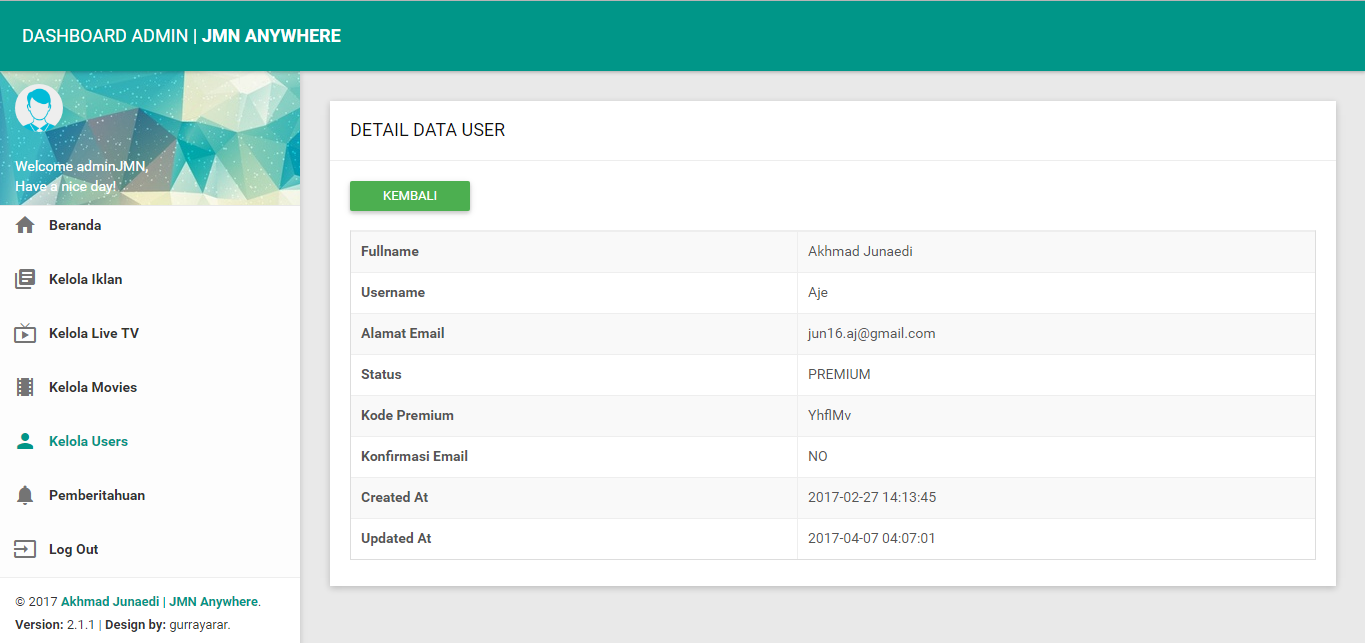
Pada halaman data *user* digunakan untuk menampilkan beberapa data *user* yang tersimpan dalam *database* yaitu *username,* alamat *email,* status, dan konfirmasi *email* serta mengelola data *user* hanya hapus data. Terdapat dua buah tombol yaitu tombol hapus yang memiliki fungsi untuk menghapus data dari *database* dan tombol *detail* untuk menampilkan data *user* secara lengkap. Pada gambar 4.24 menunjukkan tampilan dari halaman kelola *user* aplikasi JMN Anywhere pada *web*.



Gambar 4.24 Tampilan halaman kelola user aplikasi JMN Anywhere pada web

1. **Halaman *Detail User***

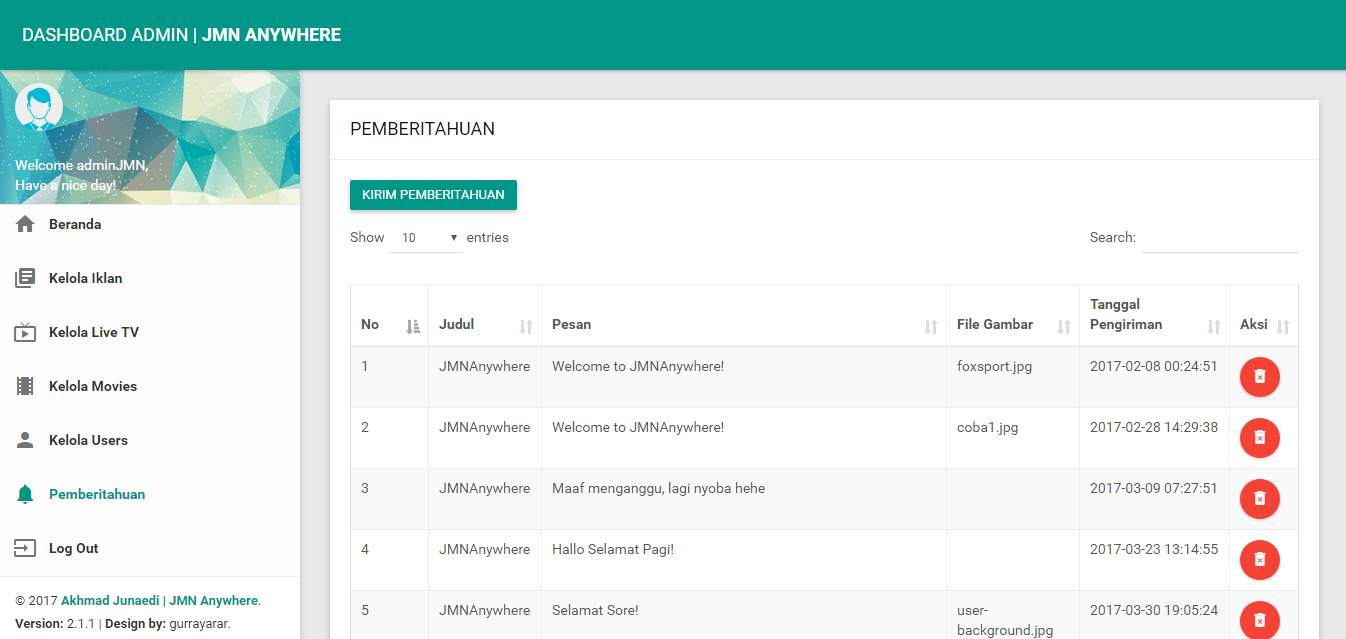
Pada halaman *detail user* digunakan untuk menampilkan data *user* secara lengkap. Data yang ditampilkan yaitu *fullname, username,* alamat *email,* status, kode *premium*, konfirmasi *email, created at,* dan *updated at*. Pada gambar 4.25 menunjukkan tampilan dari halaman *detail movies* aplikasi JMN Anywhere pada *web*.



Gambar 4.25 Tampilan halaman detail user aplikasi JMN Anywhere pada web

1. **Halaman Pemberitahuan**

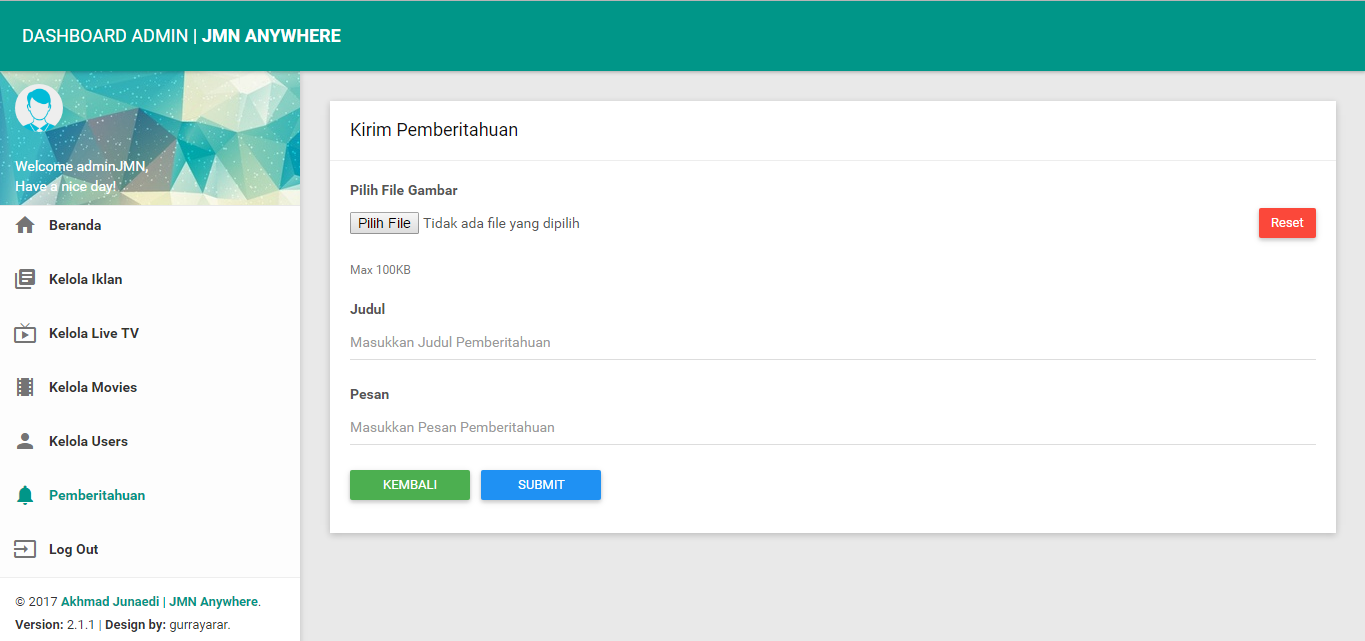
Pada halaman pemberitahuan digunakan untuk menampilkan semua data pemberitahuanyang telah dikirim dan tersimpan dalam *database* yaitu judul, pesan, *file* gambar, dan tanggal pengiriman serta mengelola data pemberitahuan hanya hapus data. Terdapat dua buah tombol yaitu tombol kirim pemberitahuan yang memiliki fungsi untuk membuka halaman kirim pemberitahuan dan tombol hapus yang memiliki fungsi untuk menghapus data dari *database*. Pada gambar 4.26 menunjukkan tampilan dari halaman pemberitahuan aplikasi JMN Anywhere pada *web*.



Gambar 4.26 Tampilan halaman pemberitahuan aplikasi JMN Anywhere pada web

1. **Halaman Kirim Pemberitahuan**

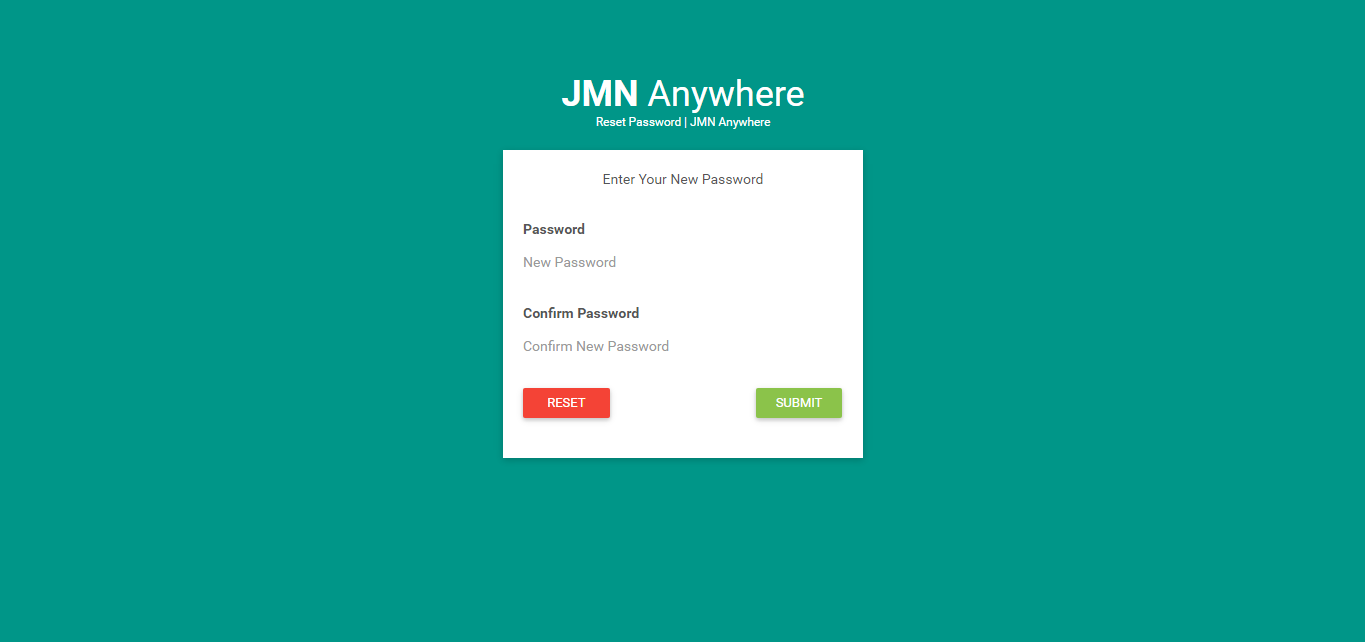
Pada halaman kirim pemberitahuan merupakan halaman yang berfungsi untuk mengirim pemberitahuan ke semua *user*. Terdapat *input form* untuk memilih *file* gambar, judul pesan, dan isi pesan. Terdapat tombol kirim yang digunakan untuk memulai pengiriman data-data pada *input form* dan menyimpan data tersebut pada *database*. Pada gambar 4.27 menunjukkan tampilan dari halaman kirim pemberitahuan aplikasi JMN Anywhere pada *web.*



Gambar 4.27 Tampilan halaman kirim pemberitahuan aplikasi JMN Anywhere pada web

1. **Halaman *Reset* *Password***

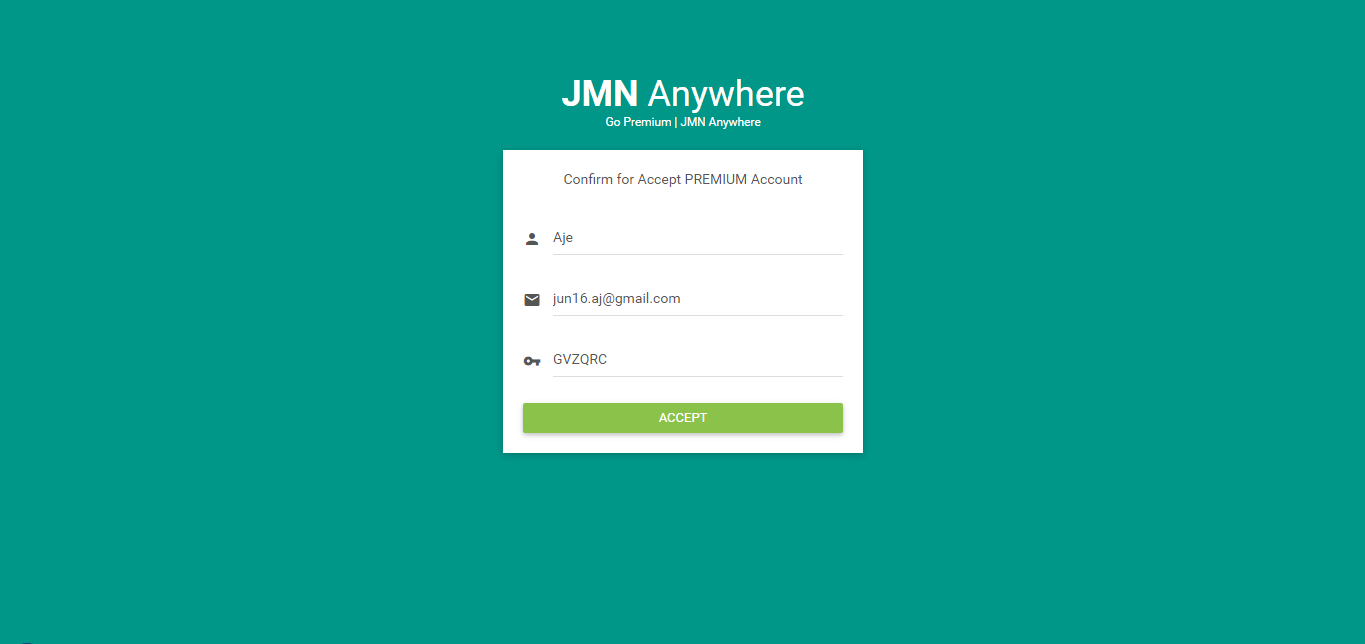
Pada halaman *reset password* merupakan halaman yang digunakan pengguna baik dari akun *free user* maupunakun *premium user* untuk memperbarui *password* jika pengguna lupa dengan *password* lama. Halaman ini hanya dapat diakses oleh pengguna melalui *link reset password* yang telah dikirim ke *email* pengguna yang melakukan permintaan *reset password*. Halaman ini terdapat dua *input form* yaitu *form password* yang digunakan untuk memasukkan *password* baru dan *form confirm password* yang digunakan untuk memasukkan konfirmasi *password* yang telah dimasukkan.Pada gambar 4.28 menunjukkan tampilan dari halaman *reset password* aplikasi JMN Anywhere pada *web.*



Gambar 4.28 Tampilan halaman reset password aplikasi JMN Anywhere pada web

1. **Halaman *Go Premium***

Pada halaman *go premium* merupakan halaman yang digunakan admin untuk menyetujui permintaan dari pengguna dengan status *free user* untuk *upgrade* ke *premium user*. Admin mengecek data dari *free user* yang ditampilkan dalam *input form* berupa *username,* alamat *email,* dan kode *premium* yang nantinya dikirim ke *email* *free user* yang melakukan permintaan. Halaman ini hanya dapat diakses oleh admin melalui *link go premium* yang telah dikirim ke *email* admin.Pada gambar 4.29 menunjukkan tampilan dari halaman *go premium* aplikasi JMN Anywhere pada *web.*



Gambar 4.29 Tampilan halaman go premium aplikasi JMN Anywhere pada web

## Pengujian

Pada tahap ini dilakukan pengujian terhadap aplikasi yang telah dibuat. Pengujian merupakan suatu tindakan yang dilakukan untuk mendapatkan informasi mengenai kualitas dari produk aplikasi yang sedang diuji. Pengujian aplikasi JMN Anywhere ini dilakukan menggunakan metode pengujian *Black-Box*. Metode pengujian *Black-Box* dilakukan bertujuan untuk menemukan kesalahan dalam beberapa kategori yaitu fungsi yang salah atau hilang, kesalahan antarmuka, kesalahan dalam struktur data atau akses *database* eksternal, perilaku atau kinerja kesalahan, dan inisialisasi dan terminasi kesalahan[10]. Tingkat keberhasilan dari pengujian ditentukan dari terpenuhinya spesifikasi kebutuhan dan skenario aplikasi.

Cara pengujian menggunakan metode pengujian *Black-Box* ini dengan mengamati jalannya aplikasi apakah sudah sesuai dengan kebutuhan atau tidak. Kemudian, hasil dari pengujian aplikasi JMN Anywhere menggunakan metode pengujian *Black-Box* disajikan dalam bentuk tabel pengujian *Black-Box* dari masing-masing fungsi, menu dan halaman yang ada dalam aplikasi JMN Anywhere.

### Pengujian Aplikasi JMN Anywhere pada Android

Pada tahap ini dilakukan pengujian semua fungsi dari aplikasi JMN Anywhere pada Android. Pengujian dilakukan pada perangkat Android yang sudah terpasang aplikasi JMN Anywhere. Berikut merupakan hasil dari pengujian *Black-Box* yang telah dilakukan.

1. **Pengujian Pada Telepon Pintar Berbasis Android**

Pada tabel 4.12 berikut ini merupakan hasil pengujian pada telepon pintar berbasis Android. Telepon pintar yang digunakan menggunakan sistem operasi dari versi 4.2 (*Jelly Bean*) sampai versi 7.1 (*Nougat*). Dari tabel 4.12 dapat dilihat bahwa semua pengujian pada telepon pintar berbasis Android, aplikasi JMN Anywhere dapat dijalankan dengan lancar.

Tabel 4.12 Tabel pengujian pada telepon pintar berbasis Android

|  |  |  |  |
| --- | --- | --- | --- |
| **Telepon Pintar** | **Spesifikasi** | **Status** | **Keterangan** |
| Smartfren Andromax U2 | * Android 4.2 * RAM 1 GB * Chipset Qualcomm Snapdragon * 540 x 960 pixels | Berhasil | Berjalan dengan lancar |
| Asus Zenfone 4 | * Android 4.4.2 * RAM 1 GB * Chipset Intel Atom Z2520 * 480 x 800 pixels | Berhasil | Berjalan dengan lancar |
| HTC One M7 | * Android 5.0.2 * RAM 2 GB * Chipset Qualcomm APQ8064T Snapdragon 600 * Full HD 1920 x 1080 pixels | Berhasil | Berjalan dengan lancar |
| Huawei P9 Lite | * Android 6.0 * RAM 3 GB * Chipset HiSilicon Kirin 650 * Full HD 1080 x 1920 pixels | Berhasil | Berjalan dengan lancar |
| Meizu M2 Note | * Android 7.1.2 * RAM 2 GB * Chipset Mediatek MT6753 * Full HD 1080 x 1920 pixels | Berhasil | Berjalan dengan lancar |

1. **Pengujian Fungsi Umum Aplikasi pada Android**

Pada tabel 4.13 berikut ini merupakan hasil pengujian pada fungsi umum dari aplikasi JMN Anywhere pada Android. Dari tabel 4.13 dapat dilihat bahwa semua pengujian fungsi umum yang ada pada aplikasi JMN Anywhere berhasil berfungsi sesuai dengan harapan.

Tabel 4.13 Tabel pengujian fungsi umum aplikasi Android

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Pengujian** | **Bentuk Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengujian** |
| Membuka aplikasi JMN Anywhere | Mengklik ikon aplikasi JMN Anywhere pada menu *drawer* perangkat Android | Tampil halaman *splash screen* dan kemudian tampil halaman *login* aplikasi JMN Anywhere | Berhasil |
| Pengujian masuk ke halaman utama aplikasi | Mengisi *form* dan menekan tombol *login* | Masuk halaman utama aplikasi | Berhasil |
| Menampilkan iklan pada *image slider* | Mengklik tab menu *home* | Tampil iklan pada *image slider* | Berhasil |
| Menampilkan halaman *about* JMN Anywhere | Mengklik tombol *about* pada *options menu* | Tampil halaman *about* JMN Anywhere | Berhasil |
| Menampilkan *toast* konfirmasi *email* | *Email* yang terdaftar tidak melakukan konfirmasi | Tampil *toast* perintah konfirmasi *email* di bagian atas | Berhasil |
| Pengujian tombol *edit profile* | Mengklik tombol *edit profile* pada tab menu *profile* | Tampil halaman *edit profile* | Berhasil |
| Pengujian tombol *change password* | Mengklik tombol *change password* pada tab menu *profile* | Tampil halaman *change password* | Berhasil |
| Mengedit *profile* pengguna | Mengubah data yang tampil pada halaman *edit profile* dan menekan tombol *checklist* pada *toolbar* | Data *profile* pengguna pada tab menu *profile* berubah | Berhasil |
| Mengganti *password* | Memasukkan *password* lama dan baru pada *input form* halaman *change password* dan menekan tombol *change password* | *Password* berhasil diganti dan disimpan pada *database* | Berhasil |
| Pengujian registrasi pengguna baru | Melengkapi *input form* pada halaman registrasi dan menekan tombol *register* | Data pengguna berhasil disimpan dan pengguna baru mendapat *email* konfirmasi akun | Berhasil |
| Menjalankan *video player* tanpa *fullscreen* | Memilih dan mengklik konten yang akan dimainkan kemudian perangkat Android dalam posisi *portrait* | *Video player* memainkan konten yang dipilih pengguna tanpa tampilan *fullscreen* | Berhasil |
| Menjalankan *video player* dengan *fullscreen* | Memilih dan mengklik konten yang akan dimainkan kemudian perangkat Android dalam posisi *landscape* | *Video player* memainkan konten yang dipilih pengguna dengan tampilan *fullscreen* | Berhasil |
| Pengujian lupa *password* | Memasukkan alamat *email* yang terdaftar pada halaman *forgot password* | Tampil *toast* berhasil mengirim *email* dan pengguna mendapat *email* yang berisi *link reset password* | Berhasil |
| Pengujian pemberitahuan tidak ada koneksi | Mematikan koneksi *mobile data* pada perangkat Android | Muncul *snackbar* dengan teks “*No connected to internet*” dan semua halaman menu muncul pemberitahuan *no internet connection* | Berhasil |
| Pengujian pemberitahuan ada koneksi | Menghidupkan koneksi *mobile data* pada perangkat Android | Muncul *snackbar* dengan teks “Connected to internet” dan menekan tombol *go online* untuk kembali menampilkan data | Berhasil |
| Pengujian *push notification* tanpa gambar | Login ke sistem aplikasi JMN Anywhere dan admin mengirim notifikasi tanpa gambar melalui *dashboard* admin pada *web* | Muncul *push notification* tanpa gambarpada perangkat Android | Berhasil |
| Pengujian *push notification* dengan gambar | Login ke sistem aplikasi JMN Anywhere dan admin mengirim notifikasi dengan gambar melalui *dashboard* admin pada *web* | Muncul *push notification* dengan gambarpada perangkat Android | Berhasil |
| Pengujian keluar dari halaman utama aplikasi | Mengklik tombol *logout* pada *options menu* | Tampil halaman *login* aplikasi | Berhasil |

1. **Pengujian Fungsi sisi *Free User***

Pada tabel 4.14 berikut ini merupakan hasil pengujian fungsi dari aplikasi JMN Anywhere yang terdapat pada sisi *free user*. Dari tabel 4.14 dapat dilihat bahwa semua pengujian fungsi yang ada pada tampilan sisi *free user* berhasil berfungsi sesuai dengan harapan.

Tabel 4.14 Tabel pengujian fungsi sisi free user

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Pengujian** | **Bentuk Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengujian** |
| Menampilkan halaman *home* | Mengklik tab menu *home* | Halaman tab menu *home* hanya menampilkan konten televisi lokal saja | Berhasil |
| Menampilkan halaman *live tv* | Mengklik tab menu *live tv* | Halaman *live tv* menampilkan data *live tv* | Berhasil |
| Menampilkan halaman *profile* | Mengklik tab menu *profile* | Halaman *profile* menampilkan data pengguna | Berhasil |
| Menampilkan halaman *more* | Mengklik tombol *more* pada halaman *live tv* dengan kategori lokal saja | Tampil halaman *more* dengan menampilkan data *live tv* dengan kategori lokal lebih lengkap | Berhasil |
| Menampilkan dialog peringatan | Mengklik konten selain televisi lokal, mengklik tombol pencarian, dan mengklik *list button* di bagian atas halaman *live tv* | Tampil dialog peringatan bahwa konten hanya untuk *premium user* | Berhasil |
| Menampilkan *snackbar* untuk *upgrade to premium* | Membuka tab menu *profile* | Muncul *snackbar* pada sisi bawah halaman *profile* | Berhasil |
| Menampilkan dialog konfirmasi *password* | Mengklik tombol *go premium* pada dialog peringatan atau mengklik pada *snackbar* *upgrade to premium* | Tampil dialog konfirmasi *password* | Berhasil |
| Pengujian permintaan *go premium* | Memasukkan konfirmasi *password* saat muncul dialog konfirmasi dan menekan tombol OK | Tampil *toast* berhasil mengirim permintaan *go premium* dan admin menerima *email* konfirmasi | Berhasil |
| Pengujian tombol *switch to premium* | Mengklik tombol *switch to premium* pada halaman *profile* | Tampil halaman *switch to premium* | Berhasil |
| Pengujian *switch to premium* | Memasukkan kode *premium* di *input form* pada halaman *switch to premium* dan menekan tombol *Go Premium* | Tampil *toast* berhasil *upgrade to premium* di bagian atas, tab menu *movies* muncul, dan halaman utama berubah | Berhasil |

1. **Pengujian Fungsi sisi *Premium User***

Pada tabel 4.15 berikut ini merupakan hasil pengujian fungsi dari aplikasi JMN Anywhere yang terdapat pada sisi *premium user*. Dari tabel 4.15 dapat dilihat bahwa semua pengujian fungsi yang ada pada tampilan sisi *premium user* berhasil berfungsi sesuai dengan harapan.

Tabel 4.15 Tabel pengujian fungsi sisi premium user

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Pengujian** | **Bentuk Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengujian** |
| Menampilkan halaman *home* | Mengklik tab menu *home* | Halaman tab menu *home* hanya menampilkan konten televisi dengan kategori lokal dan *news* serta konten *movies* dengan kategori *new movie releases* | Berhasil |
| Menampilkan halaman *live tv* | Mengklik tab menu *live tv* | Halaman *live tv* menampilkan data *live tv* | Berhasil |
| Pengujian tombol *music* | Mengklik *list button music* di bagian atas halaman *live tv* | Tampil halaman *music* dan hanya menampilkan data *live tv* dengan kategori *music* | Berhasil |
| Pengujian tombol *entertainment* | Mengklik *list button entertainment* di bagian atas halaman *live tv* | Tampil halaman *entertainment* dan hanya menampilkan data *live tv* dengan kategori *entertainment* | Berhasil |
| Pengujian tombol *knowledge* | Mengklik *list button knowledge* di bagian atas halaman *live tv* | Tampil halaman *knowledge* dan hanya menampilkan data *live tv* dengan kategori *knowledge* | Berhasil |
| Pengujian tombol *movies* | Mengklik *list button movies* di bagian atas halaman *live tv* | Tampil halaman *movies* dan hanya menampilkan data *live tv* dengan kategori *movies* | Berhasil |
| Pengujian tombol *kids* | Mengklik *list button kids* di bagian atas halaman *live tv* | Tampil halaman *kids* dan hanya menampilkan data *live tv* dengan kategori *kids* | Berhasil |
| Pengujian tombol *others* | Mengklik *list button others* di bagian atas halaman *live tv* | Tampil halaman *others* dan hanya menampilkan data *live tv* dengan kategori *others* | Berhasil |
| Menampilkan halaman *movies* | Mengklik tab menu *movies* | Halaman *movies* menampilkan data *movies* | Berhasil |
| Menampilkan halaman info *movies* | Mengklik pada konten *movies* yang dipilih | Menampilkan data secara lengkap pada *movies* yang dipilih | Berhasil |
| Memainkan *movies* pada *video player* | Mengklik tombol *play* yang ada di *options menu* pada *cardview* dan mengklik tombol *play* yang ada pada halaman info *movies* | Memainkan *movies* sesuai yang dipilih pada *video player* | Berhasil |
| Menampilkan halaman *profile* | Mengklik tab menu *profile* | Halaman *profile* menampilkan data pengguna | Berhasil |
| Menampilkan halaman *more* | Mengklik tombol *more* pada halaman *live tv* dan *movies* | Tampil halaman *more* dengan menampilkan data *live tv* dan *movies* secara lengkap sesuaidengan kategori yang dipilih | Berhasil |
| Pengujian tombol *genre* | Mengklik *list button genre* di bagian atas halaman *movies* | Menampilkan daftar *genre* | Berhasil |
| Memilih *genre movies* | Mengklik *genre* yang dipilihpada daftar *genre* | Menampilkan data *movies* sesuai dengan *genre* yang dipilih | Berhasil |
| Pengujian tombol *asian* | Mengklik *list button asian* di bagian atas halaman *movies* | Tampil halaman *asian* dan hanya menampilkan data *movies* dengan kategori *asian* | Berhasil |
| Pengujian tombol *indonesian* | Mengklik *list button indonesian* di bagian atas halaman *movies* | Tampil halaman *indonesian* dan hanya menampilkan data *movies* dengan kategori *indonesian* | Berhasil |
| Pengujian tombol *box office* | Mengklik *list button box office* di bagian atas halaman *movies* | Tampil halaman *box office* dan hanya menampilkan data *movies* dengan kategori *box office* | Berhasil |
| Pengujian tombol *search* | Mengklik tombol *search* pada *toolbar* | Tampil *input form* pada *toolbar* untuk memasukkan kata kunci | Berhasil |
| Pengujian pencarian data | Memasukkan kata kunci pada *form* pencarian*,* dan menekan tombol enter pada *keyboard* | Menampilkan data *live tv* maupun *movies* sesuai dengan kata kunci | Berhasil |
| Pengujian pemberitahuan tidak ada data yang dicari | Memasukkan kata kunci sembarang pada *form* pencarian | Menampilkan pemberitahuan berupa teks “*No results data for TV*” atau “*No results data for Movies*” | Berhasil |
| Pengujian hapus *history* data pencarian | Mengklik tombol *clear search history* pada *options menu* | *History* data pencarian berhasil dihapus | Berhasil |

### Pengujian Aplikasi *Dashboard* Admin JMN Anywhere pada *Web*

Pada tahap ini dilakukan pengujian semua fungsi dari aplikasi *dashboard* admin JMN Anywhere pada *web*. Pengujian dilakukan menggunakan *web browser* Google Chrome. Berikut merupakan hasil dari pengujian *Black-Box* yang telah dilakukan.

1. **Pengujian Fungsi Umum Aplikasi pada *Web***

Pada tabel 4.16 berikut ini merupakan hasil pengujian pada fungsi umum dari aplikasi *dashboard* admin JMN Anywhere pada *web*. Dari tabel 4.16 dapat dilihat bahwa semua pengujian fungsi umum yang ada pada aplikasi *dashboard* admin JMN Anywhere berhasil berfungsi sesuai dengan harapan.

Tabel 4.16 Tabel pengujian fungsi umum pada web

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Pengujian** | **Bentuk Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengujian** |
| Membuka aplikasi *web* | Mengakses alamat url aplikasi *web* | Tampil halaman *login* aplikasi *web* | Berhasil |
| Pengujian masuk ke halaman utama aplikasi *web* | Mengisi *form* dan menekan tombol *login* | Masuk halaman utama aplikasi *web* | Berhasil |
| Melakukan *reset password* | Mengakses alamat url *reset password* dan memasukkan *password baru* dan konfirmasi *password* | Tampil halaman *reset password* dan tampil pemberitahuan berhasil menyimpan *password* baru | Berhasil |
| Melakukan konfirmasi persetujuan *go premium* | Mengakses alamat url *go premium* yang ada di *email* dan menekan tombol *accept* | Tampil halaman konfirmasi *go premium* dan tampil tampil pemberitahuan berhasil mengirim kode *premium* ke *email* *free user* | Berhasil |
| Pengujian tombol navigasi beranda | Mengklik tombol beranda | Tampil halaman beranda dan menampilkan lima *info box* | Berhasil |
| Pengujian tombol navigasi kelola iklan | Mengklik tombol kelola iklan | Tampil halaman kelola iklan dan menampilkan data iklan | Berhasil |
| Pengujian tombol navigasi kelola *live tv* | Mengklik tombol kelola *live tv* | Tampil halaman kelola *live tv* dan menampilkan data *live tv* | Berhasil |
| Pengujian tombol navigasi kelola *movies* | Mengklik tombol kelola *movies* | Tampil halaman kelola *movies* dan menampilkan data *movies* | Berhasil |
| Pengujian tombol navigasi kelola *users* | Mengklik tombol kelola *users* | Tampil halaman kelola *users* dan menampilkan data *users* | Berhasil |
| Pengujian tombol navigasi pemberitahuan | Mengklik tombol pemberitahuan | Tampil halaman pemberitahuan dan menampilkan data pemberitahuan | Berhasil |
| Pengujian keluar dari halaman utama aplikasi *web* | mengklik tombol *logout* pada *main navigation* | Tampil halaman *login* aplikasi *web* | Berhasil |

1. **Pengujian Fungsi pada Menu Kelola Iklan**

Pada tabel 4.17 berikut ini merupakan hasil pengujian fungsi aplikasi *dashboard* admin JMN Anywhere pada *web* yang terdapat dalam menu kelola iklan. Dari tabel 4.17 dapat dilihat bahwa semua pengujian fungsi yang terdapat dalam menu kelola iklan berhasil berfungsi sesuai dengan harapan.

Tabel 4.17 Tabel pengujian fungsi pada menu kelola iklan

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Pengujian** | **Bentuk Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengujian** |
| Pengujian tombol tambah iklan baru | Mengklik tombol tambah iklan baru | Tampil halaman tambah iklan baru | Berhasil |
| Tambah iklan baru | Mengisi *form* pada halaman tambah iklan baru dan mengklik tombol *submit* | Tampil data iklan baru pada halaman kelola iklan | Berhasil |
| Pengujian tombol *edit* data iklan | Mengklik tombol *edit* iklan | Tampil halaman *edit* iklan | Berhasil |
| Mengubah data iklan | Mengubah data iklan pada *form* halaman *edit* iklan dan mengklik tombol simpan | Tampil perubahan data iklan pada halaman kelola iklan | Berhasil |
| Pengujian tombol hapus data iklan | Mengklik tombol hapus iklan | Tampil *alert dialog* konfirmasi hapus data | Berhasil |
| Menghapus data iklan | Mengklik tombol Ya, hapus sekarang! | Data iklan yang telah dihapus tidak tampil pada halaman kelola iklan | Berhasil |

1. **Pengujian Fungsi pada Menu Kelola *Live TV***

Pada tabel 4.18 berikut ini merupakan hasil pengujian fungsi aplikasi *dashboard* admin JMN Anywhere pada *web* yang terdapat dalam menu kelola *live tv*. Dari tabel 4.18 dapat dilihat bahwa semua pengujian fungsi yang terdapat dalam menu kelola *live tv* berhasil berfungsi sesuai dengan harapan.

Tabel 4.18 Tabel pengujian fungsi pada menu kelola live tv

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Pengujian** | **Bentuk Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengujian** |
| Pengujian tombol tambah *live tv* baru | Mengklik tombol tambah *live tv* baru | Tampil halaman tambah *live tv* baru | Berhasil |
| Tambah *live tv* baru | Mengisi *form* pada halaman tambah *live tv* baru dan mengklik tombol *submit* | Tampil data *live tv* baru pada halaman kelola *live tv* | Berhasil |
| Pengujian tombol *edit* data *live tv* | Mengklik tombol *edit live tv* | Tampil halaman *edit live tv* | Berhasil |
| Mengubah data *live tv* | Mengubah data iklan pada *form* halaman *edit live tv* dan mengklik tombol simpan | Tampil perubahan data *live tv* pada halaman kelola *live tv* | Berhasil |
| Pengujian tombol hapus data *live tv* | Mengklik tombol hapus *live tv* | Tampil *alert dialog* konfirmasi hapus data | Berhasil |
| Menghapus data *live tv* | Mengklik tombol Ya, hapus sekarang! | Data *live tv* yang telah dihapus tidak tampil pada halaman kelola *live tv* | Berhasil |

1. **Pengujian Fungsi pada Menu Kelola *Movies***

Pada tabel 4.19 berikut ini merupakan hasil pengujian fungsi aplikasi *dashboard* admin JMN Anywhere pada *web* yang terdapat dalam menu kelola *movies*. Dari tabel 4.19 dapat dilihat bahwa semua pengujian fungsi yang terdapat dalam menu kelola *movies* berhasil berfungsi sesuai dengan harapan.

Tabel 4.19 Tabel pengujian fungsi pada menu kelola movies

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Pengujian** | **Bentuk Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengujian** |
| Pengujian tombol tambah *movies* baru | Mengklik tombol tambah *movies* baru | Tampil halaman tambah *movies* baru | Berhasil |
| Tambah *movies* baru | Mengisi *form* pada halaman tambah *movies* baru dan mengklik tombol *submit* | Tampil data *movies* baru pada halaman kelola *movies* | Berhasil |
| Pengujian tombol *edit* data *movies* | Mengklik tombol *edit movies* | Tampil halaman *edit movies* | Berhasil |
| Mengubah data *movies* | Mengubah data iklan pada *form* halaman *edit movies* dan mengklik tombol simpan | Tampil perubahan data *movies* pada halaman kelola *movies* | Berhasil |
| Pengujian tombol *detail movies* | Mengklik tombol *detail movies* | Tampil halaman *detail movies* | Berhasil |
| Pengujian tombol hapus data *movies* | Mengklik tombol hapus *movies* | Tampil *alert dialog* konfirmasi hapus data | Berhasil |
| Menghapus data *movies* | Mengklik tombol Ya, hapus sekarang! | Data *movies* yang telah dihapus tidak tampil pada halaman kelola *movies* | Berhasil |

1. **Pengujian Fungsi pada Menu Kelola *Users***

Pada tabel 4.20 berikut ini merupakan hasil pengujian fungsi aplikasi *dashboard* admin JMN Anywhere pada *web* yang terdapat dalam menu kelola *users*. Dari tabel 4.20 dapat dilihat bahwa semua pengujian fungsi yang terdapat dalam menu kelola *users* berhasil berfungsi sesuai dengan harapan.

Tabel 4.20 Tabel pengujian fungsi pada menu kelola users

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Pengujian** | **Bentuk Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengujian** |
| Pengujian tombol *detail user* | Mengklik tombol *detail user* | Tampil halaman *detail user* | Berhasil |
| Pengujian tombol hapus data *user* | Mengklik tombol hapus *user* | Tampil *alert dialog* konfirmasi hapus data | Berhasil |
| Menghapus data *user* | Mengklik tombol Ya, hapus sekarang! | Data *user* yang telah dihapus tidak tampil pada halaman kelola *users* | Berhasil |

1. **Pengujian Fungsi pada Menu Pemberitahuan**

Pada tabel 4.21 berikut ini merupakan hasil pengujian fungsi aplikasi *dashboard* admin JMN Anywhere pada *web* yang terdapat dalam menu pemberitahuan. Dari tabel 4.21 dapat dilihat bahwa semua pengujian fungsi yang terdapat dalam menu pemberitahuan berhasil berfungsi sesuai dengan harapan.

Tabel 4.21 Tabel pengujian fungsi pada menu pemberitahuan

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Pengujian** | **Bentuk Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengujian** |
| Pengujian tombol kirim pemberitahuan | Mengklik tombol kirim pemberitahuan | Tampil halaman kirim pemberitahuan | Berhasil |
| Kirim pemberitahuan | Mengisi *form* pada halaman kirim pemberitahuan dan mengklik tombol kirim | Tampil pemberitahuan pada perangkat Android pengguna dan tampil data pemberitahuan pada halaman pemberitahuan | Berhasil |
| Pengujian tombol hapus data pemberitahuan | Mengklik tombol hapus pemberitahuan | Tampil *alert dialog* konfirmasi hapus data | Berhasil |
| Menghapus data pemberitahuan | Mengklik tombol Ya, hapus sekarang! | Data pemberitahuan yang telah dihapus tidak tampil pada halaman pemberitahuan | Berhasil |

## Pembahasan

Aplikasi JMN Anywhere pada Android ini hanya pengguna yang sudah terdaftar yang dapat menggunakan aplikasi. Pengguna dibedakan menjadi dua status pengguna yaitu *free user* dan *premium user.* Jika belum terdaftar pengguna dapat melakukan registrasi pada menu yang disediakan. Pengguna baru yang telah melakukan registrasi akan diberikan hak akses aplikasi dengan status *free user*. *Free user* memiliki hak akses hanya dapat mengakses siaran televisi lokal saja dan pada halaman utama aplikasi hanya tersedia tiga tab menu yaitu *home, live tv,* dan *profile*, sedangkan *premium user* memiliki hak akses penuh terhadap konten yang disediakan dalam aplikasi dan pada halaman utama aplikasi tersedia empat tab menu yaitu *home, live tv,* dan *profile. Free user* dapat mengubah statusnya menjadi *premium user* dengan melakukan permintaan *upgrade to premium*. Setelah melakukan permintaan *upgrade to premium, free user* akan mendapatkan kode *premium* yang dikirimkan oleh admin melalui *email*. Kode *premium* yang telah didapatkan *free user* kemudian digunakan untuk *upgrade to premium* melalui menu *switch to premium* yang terdapat pada tab menu *profile*. Setelah berhasil melakukan validasi kode *premium*, *free user* akan otomatis menjadi *premium user* dan dapat menikmati layanan *premium user.* Pengguna aplikasi dapat mengubah *profile* dan *password* pada tab menu *profile*.

Pada aplikasi *web* hanya admin yang dapat mengakses halaman utama. Pada halaman utama aplikasi *web* tersedia menu navigasi yang dapat digunakan admin untuk mengelola data seperti kelola iklan, kelola *live tv*, kelola *movies,* kelola *user,* dan kirim pemberitahuan. Selain itu, admin berperan dalam persetujuan *upgrade to premium* ketika menerima permintaan *upgrade to premium* dari pengguna berstatus *free user*.

Pengujian yang dilakukan pada aplikasi JMN Anywhere pada Android maupun pada *web* ini menggunakan metode pengujian kotak hitam atau *Black-Box*. Pengujian ini dilakukan untuk memeriksa terhadap fungsi aplikasi dan tombol-tombol yang terdapat pada aplikasi Android maupun *web*. Pengujian ini juga dilakukan untuk mengetahui apabila masih terdapat kesalahan dalam aplikasi ketika dijalankan. Pengujian pada Android dilakukan dengan menggunakan telepon pintar berbasis Android dengan sistem operasi versi 4.2 (*Jelly Bean*) sampai versi 7.1 (*Nougat*). Jika masih terdapat kesalahan akan segera diberikan solusi dan diperbaiki. Dan dalam pengujian yang telah dilakukan pada semua fungsi dan tombol aplikasi pada Android maupun *web* telah berhasil dan berfungsi sesuai dengan harapan.

# BAB V PENUTUP

## Kesimpulan

Berdasarkan penelitian yang telah dilakukan didapatkan beberapa kesimpulan sebagai berikut.

* + - 1. Aplikasi JMN Anywhere yang dikembangkan telah berhasil dirancang dan dibangun menggunakan bahasa pemrograman PHP dan Java. Aplikasi JMN Anywhere sudah dapat berjalan pada *web* untuk sisi admin dan pada Android untuk *free user* dan *premium user*.
      2. Aplikasi JMN Anywhere sudah dapat digunakan sebagai media penyiaran televisi dan *video on demand* melalui *video streaming* pada PT SIMS Jogja Medianet.
      3. Sistem aplikasi JMN Anywhere sudah dapat berjalan pada perangkat telepon pintar berbasis sistem operasi Android versi 4.2 (*Jelly Bean*)sampai 7.1 (*Nougat*).
      4. Berdasarkan hasil pengujian aplikasi menggunakan *Black-Box*, masing-masing fungsi yang ada dalam aplikasi yang dibuat ini telah berhasil dan berjalan sesuai dengan hasil yang diharapkan.

## Saran

Saran yang dapat diberikan oleh Penulis untuk penelitian selanjutnya adalah sebagai berikut.

1. Aplikasi JMN Anywhere dapat dikembangkan lebih lanjut dengan menambahkan fitur pembayaran ketika akan melakukan *upgrade* dari akun *free user* ke akun *premium user*.
2. Aplikasi JMN Anywhere dapat dikembangkan lebih lanjut dengan menambahkan fitur untuk mengunduh konten *video on demand* agar pengguna dapat menonton video kapan saja tanpa koneksi internet.
3. Aplikasi JMN Anywhere dapat dikembangkan lebih lanjut dengan menambahkan fitur pengaturan kualitas video agar pengguna dapat menyesuaikan kualitas video yang diinginkan.

# DAFTAR PUSTAKA

[1] C. Rick, “The State of Media and Entertainment Video.,” *Streaming Media*, pp. 12–18, 2014.

[2] H. Joshi, “Digital Media: Rise of On-demand Content,” *Deloitte*, 2015.

[3] A. Albasir, V. Mahinthan, K. Naik, A. Abogharaf, N. Goel, and B. J. Plourde, “Performance Testing of Mobile Applications on Smartphones,” *Int. J. Handheld Comput. Res.*, vol. 5, no. 4, pp. 36–47, 2014.

[4] F. Kozamernik, “Media Streaming over the Internet - an overview of delivery technologies,” *EBU Tech. Rev.*, no. October, pp. 1–15, 2002.

[5] D. Wu, S. Member, Y. T. Hou, and W. Zhu, “Streaming Video over the Internet : Approaches and Directions,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, 2001.

[6] CohuHD, “Video Streaming Protocols,” *White Pap.*, pp. 1–5, 2013.

[7] I. Sommerville, *Software Engineering*, 9th ed. Pearson Education, Inc, 2011.

[8] M. A. Racacho and S. I. Sur, “An Object Oriented Design Method For Database Systems And Computer Applications,” *Int. J. Adv. Res. IT Eng.*, vol. 5, no. 6, pp. 18–26, 2016.

[9] D. Bell, “An Introduction to The Unified Modeling Language,” *UML basics*, 2003. [Online]. Available: http://www.ibm.com/developerworks/rational/library/769.html. [Accessed: 18-Mar-2017].

[10] R. G. Pressman, *Software Engineering*, 7th ed. New York: The McGraw-Hill Companies, Inc, 2010.