

# Aplikasi Multimedia Untuk Karaoke Online

Riky Andreas Saputra<sup>#1</sup>, Sulaeman Santoso<sup>\*2</sup>

<sup>#</sup> Program Studi S1 Teknik Informatika, Universitas Kristen Maranatha  
Jl. Prof. Drg. Surya Sumantri 65 Bandung

<sup>1</sup>1672037@maranatha.ac.id

<sup>2</sup>sulaeman.santoso@it.maranatha.edu

**Abstract** — DeJava Karaoke is a family karaoke place in the city of Pematang Siantar. During the pandemic, DeJava Karaoke needed a solution to the cessation of business activities due to government policies. An online-based karaoke system can be the right solution, because customers can carry out karaoke activities from home. Customers can register to register as a member at a fee that can be set by DeJava Karaoke. After becoming a member, features that can be accessed are playing karaoke videos, search videos, request videos, hide vocals and make playlists. The test results on the system using the blackbox testing method prove that the system's functionality has been running according to the predetermined design. The online karaoke system is expected to be an opportunity for DeJava Karaoke to continue running business activities during a pandemic.

**Keywords** - karaoke, website karaoke, karaoke online, website karaoke online, karaoke lyric

## I. PENDAHULUAN

DeJava Karaoke merupakan salah satu tempat hiburan keluarga yang menawarkan fasilitas untuk bernyanyi dengan suasana nyaman, bersih dan modern. Saat ini, Fasilitas yang dimiliki oleh DeJava Karaoke telah dilengkapi dengan sistem sentralisasi data lagu yang dapat diakses oleh komputer yang disediakan di setiap ruangan.

Pada era pandemi, DeJava Karaoke mengalami hambatan bisnis karena peraturan Pemerintah untuk menghentikan kegiatan usaha yang dapat menjadi penyebab penyebaran Virus Covid-19. Dengan penghentian kegiatan usaha, maka DeJava karaoke tidak mendapatkan pendapatan. Sebagai solusi, agar tetap dapat menjalankan kegiatan usaha, maka pada penelitian ini mengusulkan Sistem Karaoke Online.

Sistem karaoke online yang diusulkan pada penelitian ini memberikan fitur simpan lagu favorit, membuat *playlist* dan informasi lagu populer atau informasi lagu terbaru. Sebagai sumber pendapatan bagi DeJava Karaoke, maka pengguna yang dapat menggunakan fasilitas karaoke wajib melakukan *registrasi* dan melakukan pembayaran biaya langganan terlebih dahulu atau pengguna yang tidak berlangganan tetap sebelum memulai lagu akan ditayangkan video berupa iklan.

Dengan sistem karaoke *online* yang diusulkan pada penelitian ini, diharapkan dapat menjadi solusi bagi DeJava Karaoke untuk tetap mendapatkan pendapatan di masa pandemi. Nilai lebih dari sistem karaoke online adalah belum banyak diterapkan oleh penyedia jasa karaoke lainnya. Hal ini menjadi keunggulan dan peluang bagi DeJava Karaoke untuk mendapatkan pelanggan.

Berdasarkan latar belakang yang telah dideskripsikan di atas, maka tujuan pembuatan website sebagai berikut:

- Menghasilkan sistem karaoke *online* di DeJava Karaoke
- Menyediakan sistem yang dapat menjadi sumber pendapatan DeJava Karaoke di masa pandemi.

## II. KAJIAN TEORI

Karaoke merupakan salah satu alternatif hiburan yang banyak diminati oleh masyarakat. Karaoke didefinisikan sebagai sebuah bentuk hiburan dimana seseorang bernyanyi dan diiringi musik dan teks lirik yang ditujukan pada suatu layar. Layanan karaoke mencakup beberapa komponen yang saling berintegrasi, yaitu: video, suara, dan teks. Video merupakan sarana visual yang memudahkan seseorang saat berkaraoke dan teks dalam video karaoke merupakan lirik lagu yang berjalan sesuai nada lagu yang didengarkan. Pada perancangan *website* karaoke online, digunakan beberapa komponen seperti: React JS sebagai *frontend* untuk UI, Node JS sebagai *backend*, MongoDB sebagai *database*, *Google Cloud Storage* sebagai *storage* penyimpanan data video, API lalal.ai sebagai penghilang vokal lagu.

Beberapa penelitian yang telah dilakukan dan memiliki kesamaan dengan penelitian ini adalah sebagai berikut:

- Perancangan dan Pembangunan Aplikasi Karaoke Online Menggunakan VoIP [1].

Penelitian ini menghasilkan aplikasi karaoke *online* dengan memanfaatkan teknologi *Voice Over Internet Protocol* (VoIP). Teknologi VoIP memungkinkan komunikasi suara jarak jauh melalui media internet dengan perangkat komputer yang mempunyai kartu suara dan perangkat *microphone*. Aplikasi yang dibuat berjalan di *web browser*, karena aplikasi dibuat dengan menggunakan bahasa pemrograman PHP 5.0.

- Perancangan dan Implementasi Video *Streaming* Berbasis Html 5 Pada Mesin Virtual [2].  
Penelitian ini mengimplementasikan video *streaming* dan layanan tambahan ftp server pada mesin virtual. Agar tujuan tersebut dapat dicapai, maka akan dibangun server dengan *virtualisasi* untuk pengimplementasian website dan FTP server. Hasil dari kegiatan penelitian ini berhasil menghasilkan sistem berbasis web yang dapat menjalankan video secara *online* dengan menggunakan jaringan internet.
- *Accompaniment separation and karaoke application based on automatic melody transcription* [3].  
Penelitian ini menghasilkan sistem karaoke dengan fitur teks transkrip secara otomatis. Hasil penelitian ini berhasil menghasilkan sistem yang dapat digunakan oleh pengguna untuk bernyanyi secara *online*. Sistem yang dihasilkan mempunyai fitur *setting* nada suara secara *real time*, sehingga menghasilkan nada suara pengguna yang baik.

Aplikasi merupakan suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer. Aplikasi juga diartikan sebagai penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan atau sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu [4]. Aplikasi *software* yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 (dua) yaitu:

- Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
- Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

Dari kedua pengertian di atas dapat disimpulkan bahwa aplikasi adalah sekumpulan perintah atau kode yang disusun secara sistematis untuk menjalankan suatu perintah yang diberikan oleh manusia melalui komponen atau *hardware* komputer yang digunakan oleh manusia dalam menjalankan program aplikasi, dengan demikian bisa membantu manusia untuk memberikan solusi dari apa yang diinginkan.

#### A. React JS

React JS merupakan sebuah *library JavaScript* bersifat *open source* yang digunakan untuk membangun *User Interface* (UI) [5]. Pengguna *React* pada penelitian ini memungkinkan pengembangan aplikasi berbasis web besar dan kompleks yang dapat mengubah datanya tanpa *refresh* halaman berikutnya. *React* menggunakan konsep modular, sehingga pembuat sistem dapat menulis kode-kode dengan skala lebih kecil untuk disatukan menjadi aplikasi utuh, dan dapat digunakan kembali (*reusable*).

#### B. Node JS

Node.js adalah perangkat lunak yang didesain untuk mengembangkan aplikasi berbasis *web* dan ditulis dalam *sintaks* bahasa pemrograman *JavaScript*. Node.js diciptakan untuk melengkapi peran *JavaScript* sebagai bahasa pemrograman yang berjalan di sisi server. Node.js memiliki fitur *built-in* HTTP server *library* yang menjadikannya mampu menjadi sebuah *web server* tanpa bantuan *software* lainnya seperti *Apache* dan *Nginx* [6]. Hal tersebut menjadikan Node.JS menjadi sebuah *runtime environment* berfungsi untuk mengeksekusi, menjalankan, dan mengimplementasikan fungsi-fungsi serta cara kerja inti dari suatu bahasa pemrograman. Sedangkan *script library* adalah kumpulan, kompilasi, atau bank data berisi skrip/ kode-kode pemrograman.

#### C. MongoDB

MongoDB adalah sebuah dokumen *oriented database* bersifat *open source*. MongoDB merupakan salah satu *database NoSQL* yang memiliki sebuah konsep penyimpanan data non-relational. Istilah NoSQL merupakan kepanjangan dari "Not Only SQL" yaitu sistem manajemen *database* berbeda dari sistem manajemen *database* rasional dalam beberapa cara [7]. Penyimpanan data tanpa perlu adanya tabel skema dan tidak ada bahasa SQL yang terlibat dalam pemakaian *database*.

MongoDB tidak mengenal adanya tabel, kolom dan baris jadi tidak ada skema dalam MongoDB (*schema-less*). Unit terkecil pada dari MongoDB adalah *document*, sedangkan kumpulan dari dokumen adalah *collection*. Seperti halnya dalam *database* rasional, *document*, ibarat sebuah *record* dan *collection* pada sebuah tabel *document* dalam MongoDB dapat memiliki atribut berbeda-beda dengan *document* lainnya walaupun pada satu *collection*. MongoDB dibuat menggunakan C++, yang memiliki beberapa fitur, antara lain: *Document-oriented storage, full index, support replication* dan *high availability, auto-sharding, querying, fast in-place updates, map reduce, gridFS, commercial support*.

Pada MongoDB tidak membutuhkan struktur tabel jadi untuk melakukan proses *insert* maka MongoDB akan otomatis membuatkan struktur tabelnya. MongoDB mempunyai performa 4x lebih cepat dibandingkan dengan MySQL. MongoDB

dikembangkan sejak Oktober 2007 oleh 10Gen dan dirilis ke publik sejak Februari 2009 yang mempunyai lisensi GNU AGPL 3.0 dan Apache License untuk drivernya.

#### *D. Google Cloud Platform (GCP)*

*Google Cloud Platform* merupakan layanan dari google terdiri dari empat *component* untuk membangun aplikasi berbasis *cloud*. Layanan yang diberikan oleh *google* antara lain: *Google AppEngine*, *Google Cloud Storage*, *Google AppEngine* dan *Google BigQuery* [8]. Pada penelitian ini menggunakan GCS (*Google Cloud Storage*) sebagai penyimpanan data.

#### *E. Lalal.ai*

Lala.ai merupakan sebuah *AI Machine Learning* untuk menghilangkan *vocal*. Lalal.ai adalah layanan pemisahan musik generasi berikutnya untuk ekstraksi cepat dan tepat. Ekstraktor pada dasarnya membantu menghilangkan trek vokal dan instrumental dengan mudah dan tanpa kehilangan kualitas. Penggunaan Lalal.ai hanya perlu menghubungkan API yang disediakan [9].

#### *F. Blackbox Testing*

Black-Box Testing merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program [10]. *Black box testing* menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

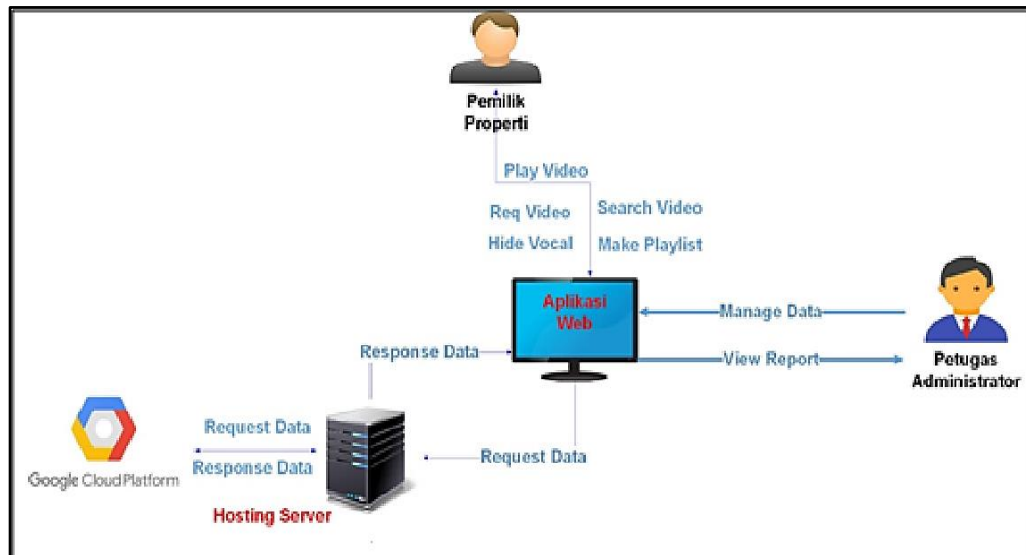
Pengujian *blackbox* dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian black box testing harus dibuat dengan kasus benar dan kasus salah. Teknik pengujian black box memungkinkan memperoleh serangkaian kondisi masukan yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Beberapa jenis kesalahan yang dapat diidentifikasi adalah fungsi tidak benar atau hilang, kesalahan antar muka, kesalahan pada struktur data (pengaksesan basis data), kesalahan performansi, kesalahan inisialisasi dan akhir program.

### III. ANALISIS DAN RANCANGAN SISTEM

Berikut ini adalah penguraian dari sistem informasi utuh ke dalam bagian-bagian dengan maksud mengidentifikasi dan mengevaluasi permasalahan-permasalahan dan hambatan-hambatan yang terjadi dengan tujuan untuk mencari kelemahan dari sistem yang berjalan.

#### *G. Diagram Arsitektur*

Diagram arsitektur menggambarkan struktur sistem secara umum terdiri dari: sistem, pengguna, dan obyek lain di luar sistem yang berperan dalam fungsional sistem. Berikut ini desain diagram arsitektur sistem usulan.

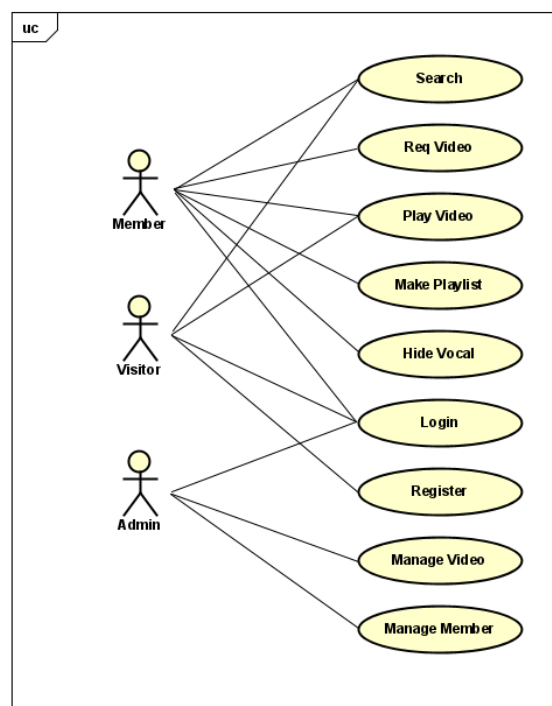


Gambar 1. Diagram Arsitektur

Pada gambar 1 terlihat sistem dapat diakses oleh pengguna melalui aplikasi *browser*. Setiap fungsional yang diakses melalui aplikasi *web* terjadi *response* dan *request* data ke *hosting* server untuk menjalankan proses fungsional yang diminta oleh pengguna. *Hosting* server mengakses data pada *database* yang disimpan di *google storage* yang merupakan bagian dari *google cloud platform*.

#### H. Use Case Diagram

Diagram *use case* menggambarkan interaksi pengguna terhadap sistem. Pada *use case diagram*, sistem usulan terdapat 3 aktor atau pengguna, yaitu: *member*, *visitor*, dan *admin*. Berikut ini desain *use case diagram* sistem usulan.

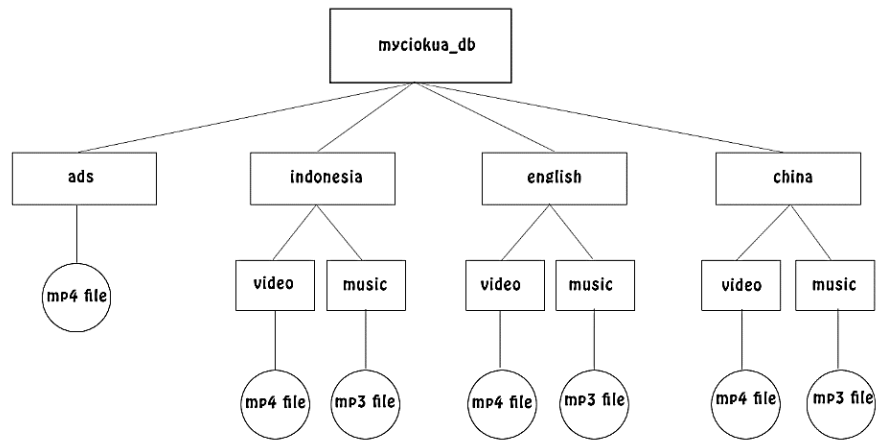


Gambar 2. Use Case Diagram

Pada Gambar 2 menjelaskan tentang *use case diagram* pada sistem, *user member* memiliki hak akses ke semua fitur kecuali *add new video*, sedangkan *visitor* (pengunjung) hanya dapat mencari, memainkan video dan menghapus vokal pada lagu, dan admin memiliki fitur tambahan, yaitu dapat menambahkan video baru pada sistem dan dapat melakukan *delete update*.

I. Tree Diagram Struktur Cloud Storage

Tree Diagram menggambarkan struktur *data cloud storage*.



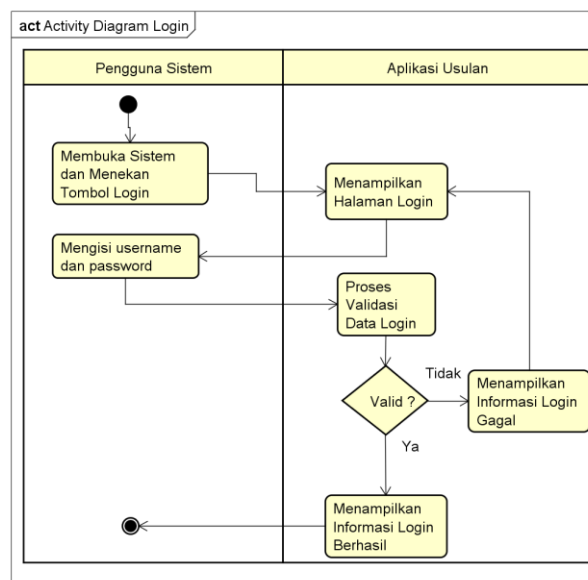
Gambar 3. Tree Diagram

Gambar 3 menjelaskan struktur tabel pada *database*, pada *database myciokua\_db* memiliki folder *ads* berisikan semua video ads dengan format mp4, folder Indonesia, English, dan China, merupakan folder lagu berdasarkan negara dan mempunyai folder tambahan, yaitu video berisikan file mp4 dan musik berisikan file mp3 lagu tanpa vokal.

J. Activity Diagram

Diagram aktifitas menggambarkan alur dari setiap *use case*. Berikut ini desain diagram aktifitas dari masing-masing *use case* yang telah dibuat.

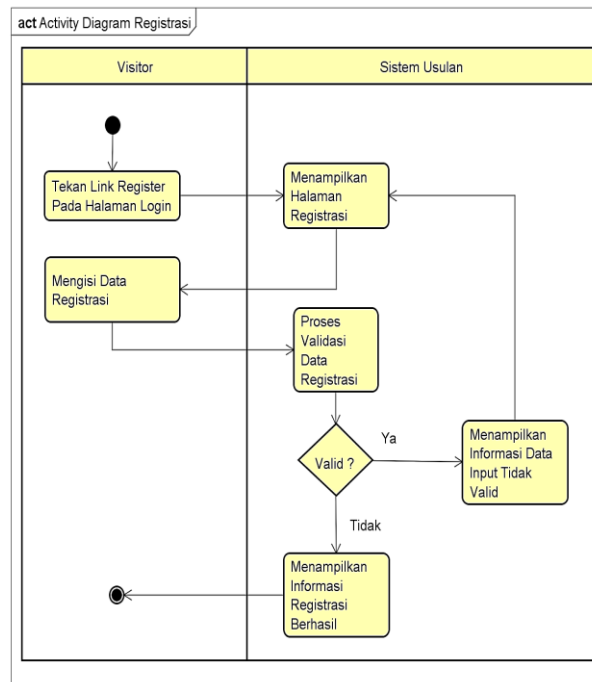
1) Activity Diagram Login:



Gambar 4. Activity Diagram Login

Gambar 4 menjelaskan diagram login dimana ketika user menekan tombol login maka sistem akan menampilkan form login dan setelah user berhasil mengisi form maka sistem akan melakukan pengecekan, jika gagal maka user diminta mengisi ulang form login dan jika benar maka sistem menampilkan tanda berhasil.

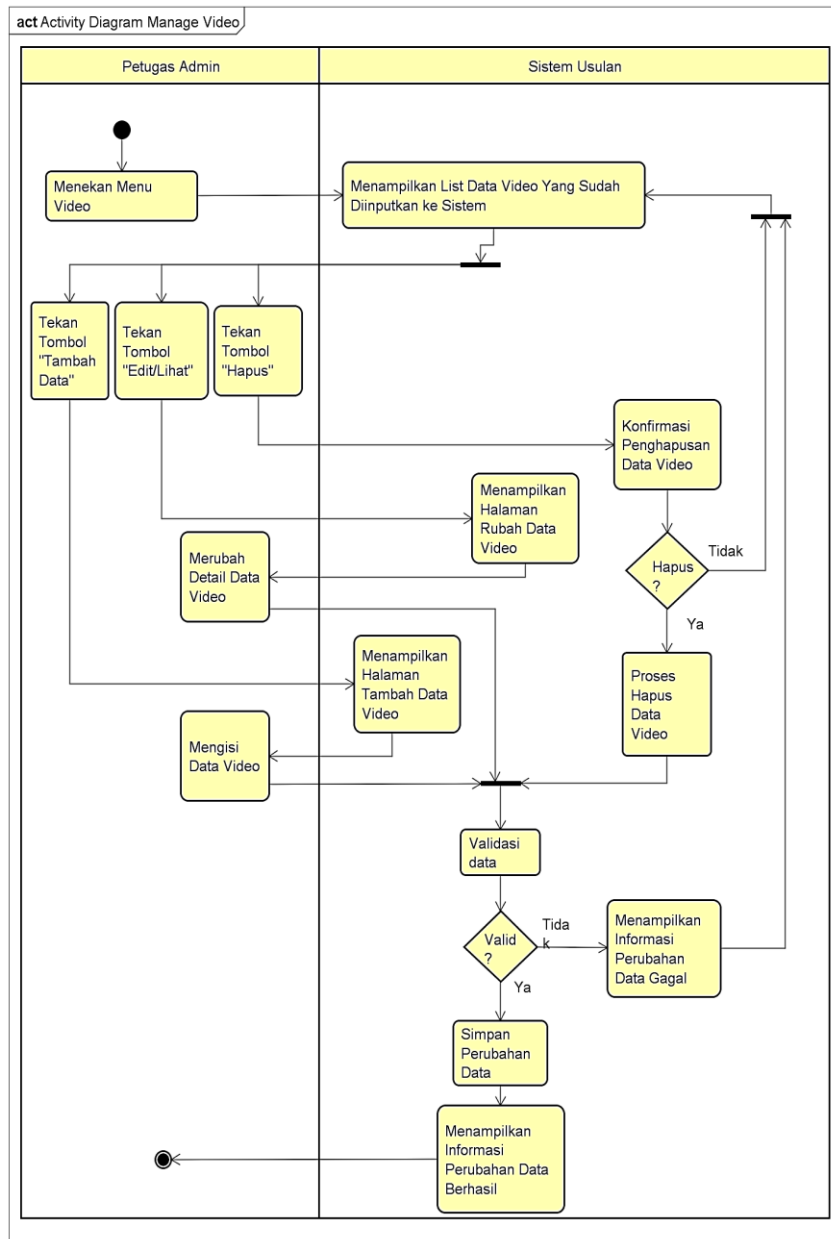
2) *Activity Diagram Register*



Gambar 5. Activity Diagram Register

Gambar 5 menjelaskan diagram register dimana ketika user menekan tombol register maka sistem akan menampilkan form register dan setelah user berhasil mengisi form maka sistem akan melakukan pengecekan, jika gagal maka user diminta mengisi ulang form register dan jika benar maka sistem menampilkan tanda berhasil.

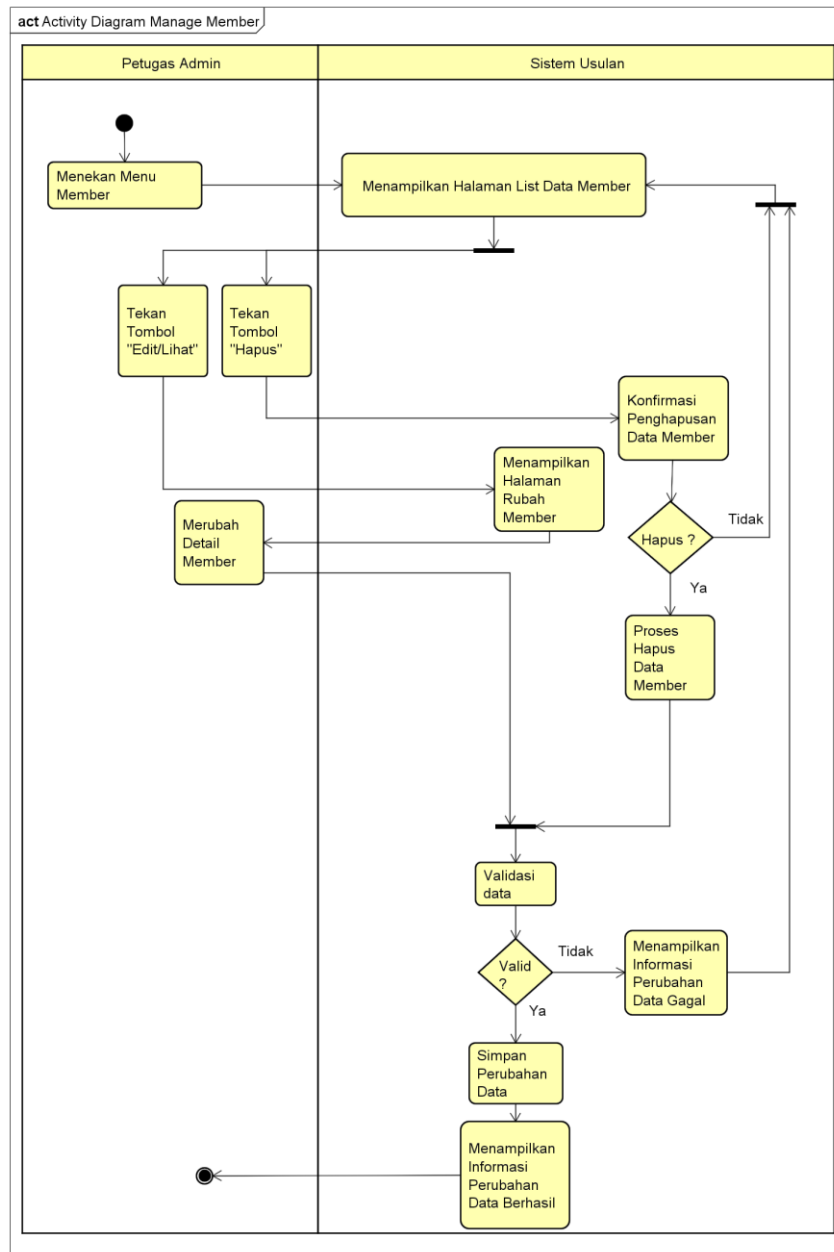
3) Activity Diagram Manage Video



Gambar 6. Activity Diagram Manage Video

Gambar 6 menjelaskan diagram *manage video* dimana ketika admin melakukan edit, *delete* & tambah data, maka sistem akan melakukan pengecekan, jika gagal maka sistem akan menampilkan gagal dan jika berhasil sistem akan menampilkan informasi berhasil.

4) Activity Diagram Manage Member

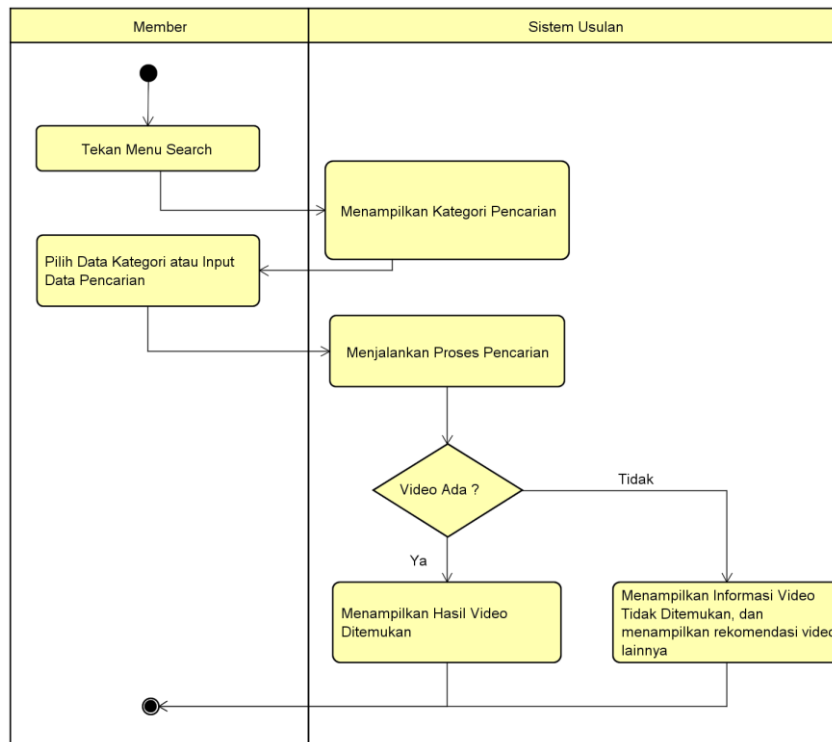


Gambar 7. Activity Diagram Manage Member

Gambar 7 menjelaskan diagram *manage member* dimana ketika admin melakukan edit atau hapus maka sistem akan melakukan pengecekan, jika gagal maka sistem akan menampilkan gagal dan jika berhasil sistem akan menampilkan informasi berhasil.



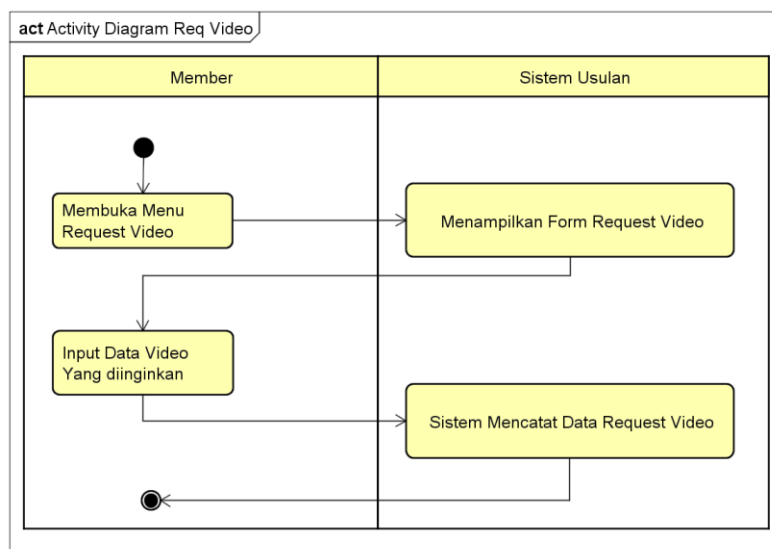
5) Activity Diagram Search Video



Gambar 8. Activity Diagram Search Video

Gambar 8 menjelaskan diagram *search video* dimana ketika member mengisi pada input pencarian maka sistem akan menampilkan video berdasarkan judul atau penyanyi dari yang dicari.

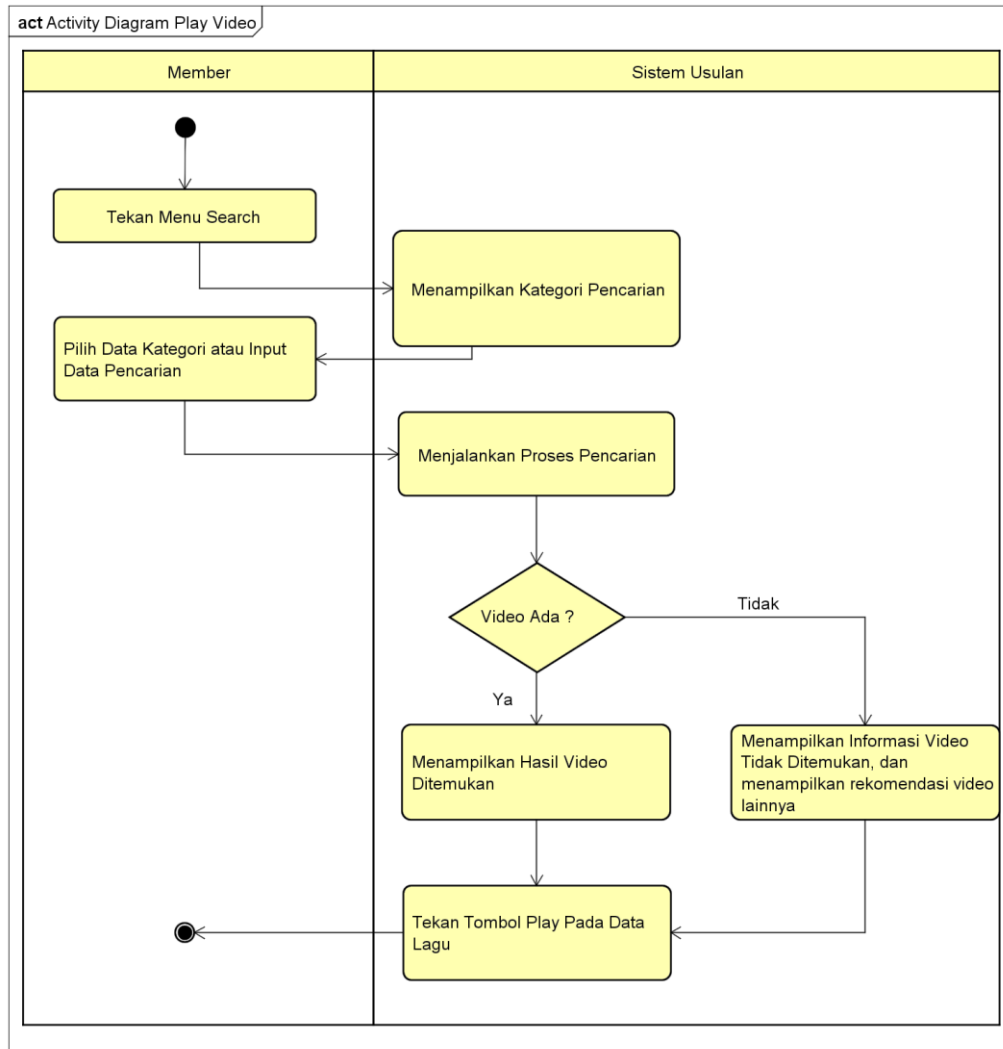
6) Activity Diagram Request Video



Gambar 9. Activity Diagram Request Video

Gambar 9 menjelaskan diagram request video dimana ketika member menekan tombol *request* maka sistem akan menampilkan form request dan setelah member berhasil mengisi form maka sistem akan menampilkan informasi berhasil dan sistem mencatat data *request video* ke dalam database.

7) Activity Diagram Play Video



Gambar 10. Activity Diagram Play Video

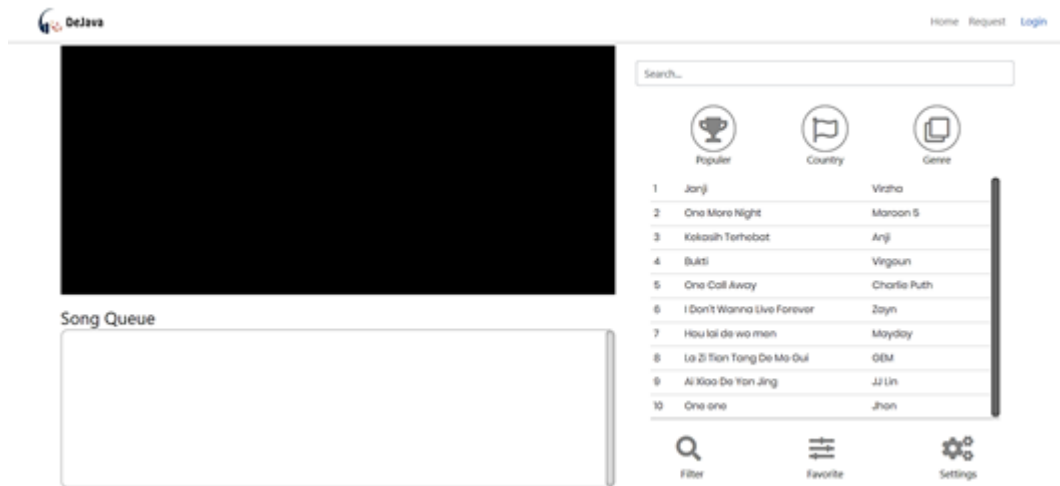
Gambar 10 menjelaskan diagram *play video* dimana ketika member mengisi menu pencarian maka sistem akan menjalankan proses pencarian jika video ada maka akan ditambahkan langsung ke daftar lagu yang dinyanyikan.

#### IV. IMPLEMENTASI

Pada bagian ini akan membahas langkah-langkah atau panduan website yang digunakan untuk mendasari penyelesaian tugas akhir.

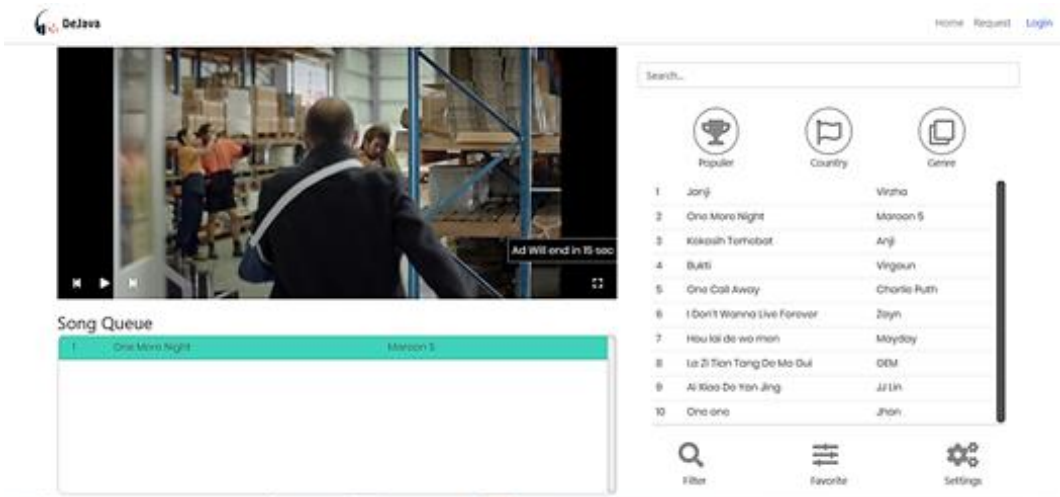
##### A. Tampilan Awal Website

Ini merupakan tampilan awal ketika mengakses website <https://dejava.herokuapp.com>.



Gambar 11. Hasil Implementasi Halaman Visitor

Ketika mengakses website <https://dejava.herokuapp.com> user akan terhitung sebagai *visitor*, pada tampilan *visitor* ketika user menekan tombol *request* secara otomatis user akan diarahkan ke halaman *login*, user juga dapat melakukan penambahan lagu pada *Song Queue*, mencari lagu berdasarkan judul atau penyanyi dan tersedianya pilihan lagu paling populer atau paling banyak dibuka dan pilihan lagu berdasarkan negara dan genre.

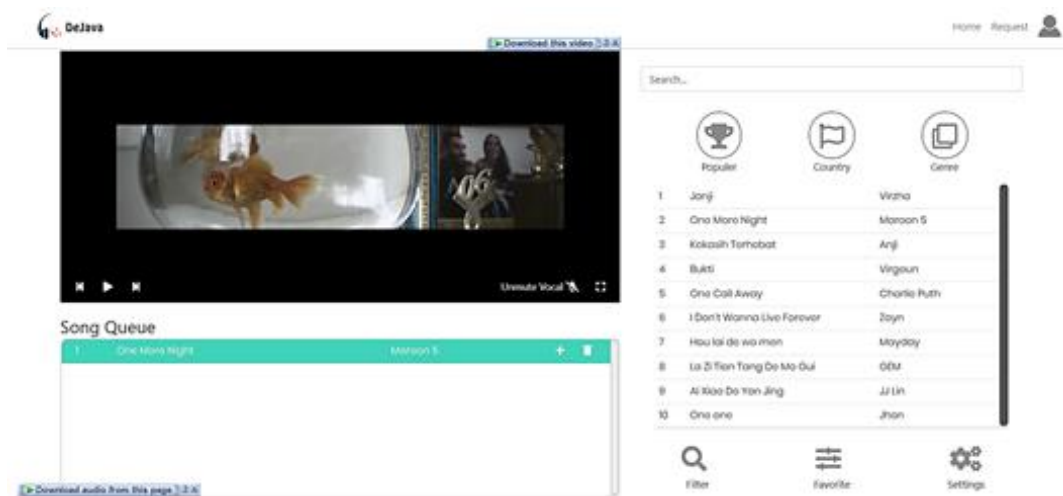


Gambar 12. Hasil Implementasi Halaman Song Queue

Gambar 12 merupakan tampilan ketika *user* telah menambah lagu ke dalam *song queue*. Pada tampilan ini *user* memiliki tambahan fitur untuk menghilangkan *vocal* dan dapat menghapus lagu dari *song queue*.

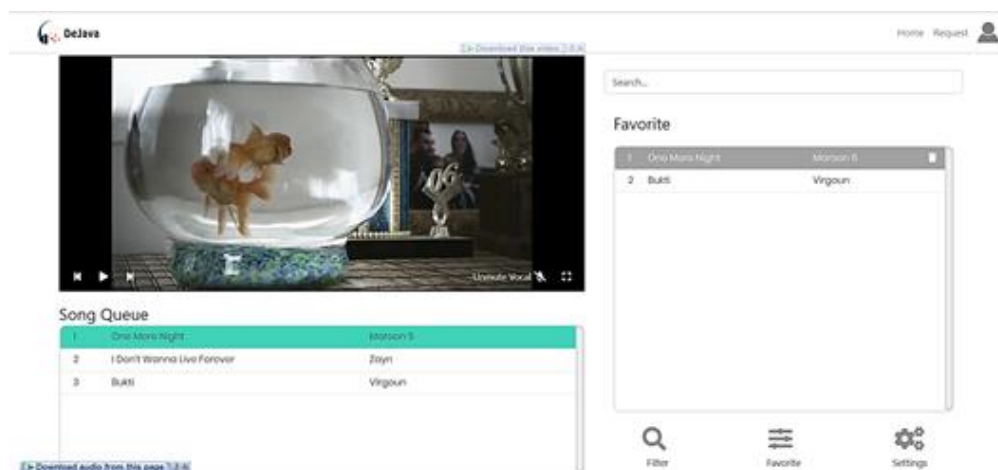
### B. Tampilan User Member

Ini merupakan tampilan *user* yang telah berhasil *login* ketika mengakses halaman *home*.



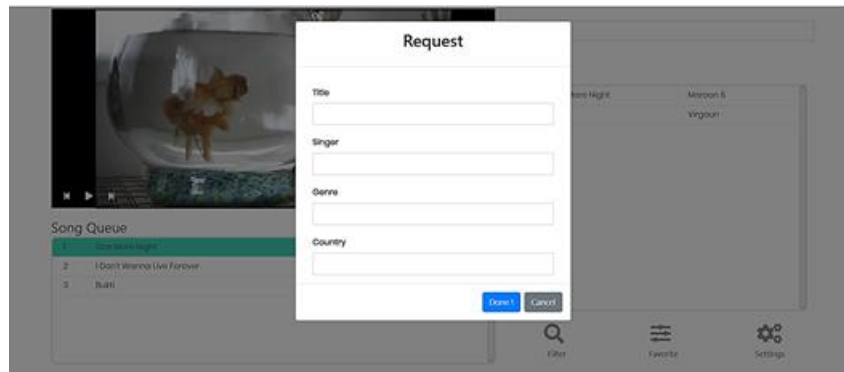
Gambar 13. Hasil Implementasi Halaman Member

Pada gambar 13, ada penambahan fitur pada dimana *user* dapat menambah lagu ke dalam daftar *favorite*. Berikut merupakan tampilan *favorite*, pada daftar *favorite* terdapat tambahan fitur yaitu *user* dapat menghapus dari daftar *favorite*.



Gambar 14. Hasil Implementasi Halaman Visitor

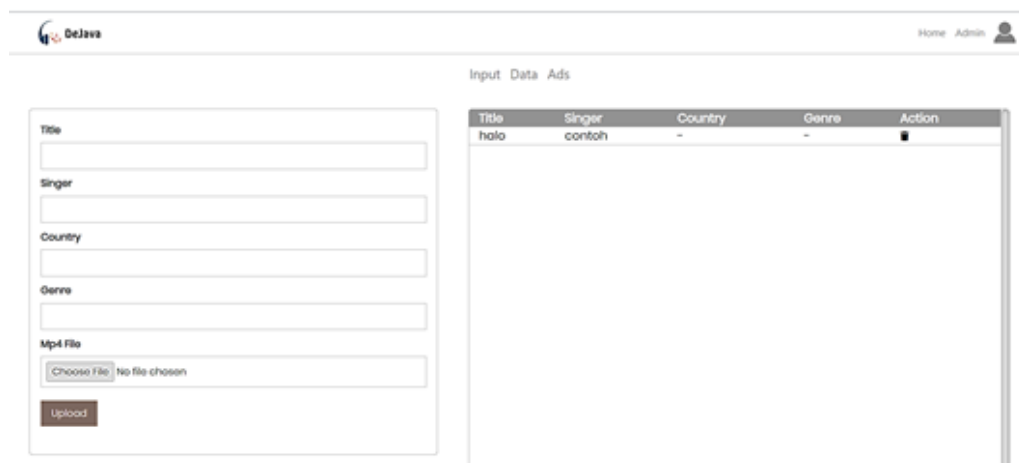
Berikut merupakan tampilan *request* yang hanya dapat diakses oleh user yang telah berhasil login, jika user mau melakukan *request* wajib memasukkan salah satu diantara *title* atau *singer* dan juga wajib memasukkan *country*.



Gambar 15. Hasil Implementasi Halaman Request Video

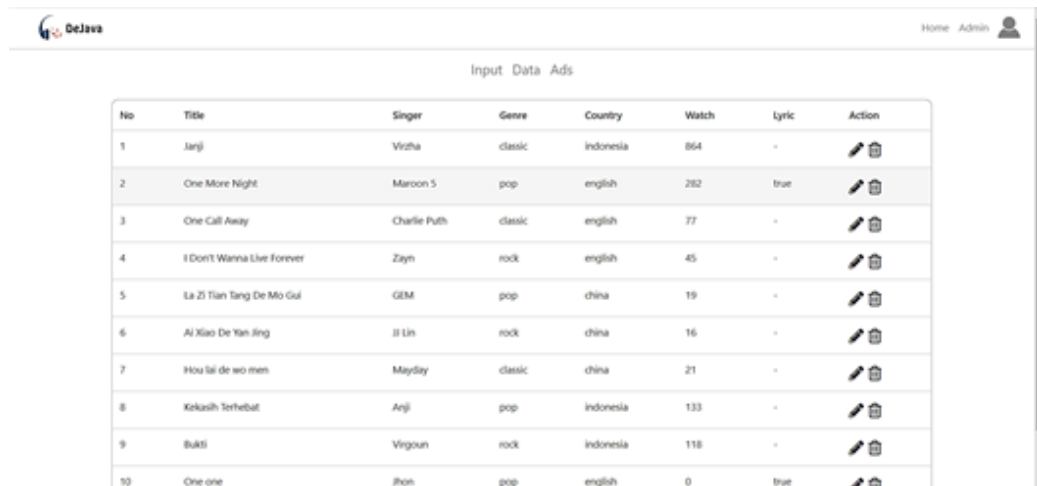
### C. Tampilan Admin





















Berikut merupakan tampilan jika *user login* sebagai *admin*, pada bagian *home* sama saja seperti tampilan sebagai *user member*, akan tetapi terdapat tambahan *page admin* yang hanya dapat dibuka jika *user role* adalah *admin*.



Gambar 16. Hasil Implementasi Halaman Manage Video

Pada *page admin* terdapat tampilan *Input Admin* dan *Data lagu*, data asd. Gambar 16 merupakan tampilan *Input Admin*. Pada tampilan ini di bagian kiri, admin dapat menambahkan lagu dimana terdapat *form* berisikan *title*, *singer*, *genre*, *file video* dan *file lyric*. Di bagian kanan terdapat daftar *request* lagu yang telah dimasukkan oleh *user member*.



No	Title	Singer	Genre	Country	Watch	Lyric	Action
1	Janji	Virzha	classic	Indonesia	864	-	 
2	One More Night	Maroon 5	pop	english	282	true	 
3	One Call Away	Charlie Puth	classic	english	77	-	 
4	I Don't Wanna Live Forever	Zayn	rock	english	45	-	 
5	La Zi Tian Tang De Mo Gui	GEM	pop	china	19	-	 
6	Al Xiao De Yan Jing	Ji Lin	rock	china	16	-	 
7	Hou lai de wo men	Mayday	classic	china	21	-	 
8	Kekasih Terhebat	Anji	pop	Indonesia	133	-	 
9	Bukti	Virgoun	rock	Indonesia	118	-	 
10	One one	Jhon	pop	english	0	true	 

Gambar 17. Hasil Implementasi Halaman Manage Video

Gambar 17 merupakan tampilan *Data Video*, pada bagian ini akan menampilkan seluruh lagu dari *database* yang tersusun dalam *table* dan setiap lagu dapat diedit dan dihapus.



Virzha - Janji

Title:

Singer:

Genre:

Country:

Watch:

Lyric:

Gambar 18. Hasil Implementasi Halaman Edit Video

Gambar 18 merupakan tampilan jika admin ingin mengedit video. Petugas admin dapat melakukan perubahan detail video lagu.

## V. SIMPULAN DAN SARAN

### A. Simpulan

Webside *dejava.herokuapp.com* berhasil menggambarkan aplikasi multimedia untuk karaoke online dimana pada penelitian ini berhasil menerapkan API dari *lalal.ai* untuk penghilang *vocal* dan menyimpan file ke dalam *cloud storage*.

Webside *dejava.herokuapp.com* diharapkan bisa menjadi sumber pendapatan bagi *Dejava Karaoke* dimasa pandemi ini dan dapat membantu pengunjung *webside* untuk melihat daftar lagu yang ingin dinyanyikan.

**B. Saran**

Berikut adalah beberapa saran yang dapat dikembangkan pada penelitian selanjutnya: (1) Browser yang *support* masih sedikit; (2) Tidak dapat mengubah posisi pada daftar *playlist*.

DAFTAR PUSTAKA

- [1] M. Yunike, "Perancangan dan Pembangunan Aplikasi Karaoke Online Menggunakan VoIP," *Seminar Nasional Informatika*, no. 1979-2328, pp. 335-341, 2008.
- [2] M. Fahrurrozi, "Perancangan dan Implementasi Video Streaming," *Fakultas Komunikasi Dan Informatika Universitas Muhammadiyah Surakarta*, 2014.
- [3] M. Ryyänen, "Accompaniment separation and karaoke application based on automatic melody transcription," *IEEE International Conference on Multimedia and Expo*, pp. 1417-1420, 2008.
- [4] A. Kadir, *Dasar Pemrograman WEB Dinamis Menggunakan PHP*, Yogyakarta : Andi, 2017 .
- [5] A. Gregorius, *Buku Pintar HTML5 + CSS3 + DreamWeaver CS6*, Yogyakarta: Jubilee Enterprise, 2015.
- [6] H. A. Puspitosari, *Pemrograman Web Database dengan PHP dan MySQL Tingkat Lanjut*, Malang: Skripta, 2017.
- [7] A. W. R. E. a. J. Sentosa, "Perbandingan Kinerja Data Manipulation," *Informatika*, p. 155–159, 2013.
- [8] N. Safaat, *Pemograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*, Bandung: Informatika, 2017.
- [9] A. Ahmad, " Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning," *Jurnal Teknologi Indonesia*, 2017.
- [10] d. M. A.S Rosa, *Rekayasa Perangkat Lunak Struktur dan 2014*, Bandung: Informatika, 2014.