
SISTEM REKOMENDASI FILM MENGGUNAKAN METODE *K-MEANS* *CLUSTERING* BERBASIS WEB

Parlaungan

Program Studi Teknik Informatika
Universitas Indraprasta PGRI
Email: parlaungan026@gmail.com

Fanisya Alva Mustika

Program Studi Teknik Informatika
Universitas Indraprasta PGRI
Email: alva.mustika@gmail.com

Harry Dhika

Program Studi Teknik Informatika
Universitas Indraprasta PGRI
Email: dhikatr@yahoo.com

ABSTRAK

Penelitian dilakukan untuk membangun sistem *website* yang dapat membantu pengguna dalam mencari film-film dengan kombinasi lebih dari satu kriteria film. Juga membantu pengguna dalam mencari preferensi genre yang pengguna inginkan dan mengelompokkan film-film tersebut. Metode penelitian yang digunakan adalah *K-Means Clustering*. Metode ini mempartisi data atau objek ke dalam *cluster* atau kelompok-kelompok. Data yang mempunyai ciri yang sama akan dikelompokkan dan dimasukkan ke dalam kelompok yang sama juga. Setelah dibuatnya sistem rekomendasi film menggunakan metode *k-means clustering* berbasis web, dilakukan uji coba sistem rekomendasi *website* menggunakan perbandingan uji coba dengan perangkat lunak *rapidminer* lalu pengujian dengan *website*. Hasil dari uji coba menunjukkan hasil data pengelompokkan yang sama antara *rapidminer* dengan *website* yang dibuat. Maka dapat disimpulkan bahwa sistem dapat mengelompokkan film dengan kemiripan tinggi berdasarkan kriteria *rating*, popularitas, dan keuntungan. Pengguna dapat memilih satu atau lebih genre juga batas tahun yang diinginkan. Sistem rekomendasi juga dapat digunakan lewat berbagai *device* karena berbasis web sehingga sistem ini dapat dengan mudah digunakan dimana saja.

Kata kunci: sistem rekomendasi, sistem rekomendasi film, metode *k-means clustering*, web.

ABSTRACT

The research was conducted to build a website system that can assist users in searching for films with a combination of more than one film criteria. Also helps users in searching for the genre preferences that users want and grouping the films. The research method used is K-Means Clustering. This method partitions data or objects into clusters or groups. Data that has the same characteristics will be grouped and put into the same group as well. After making a web-based film recommendation system using k-means clustering method, a trial of the website recommendation system was carried out using a trial comparison between rapidminer software and the recommendation system on the website. The results of the trial show the results of grouping the same data between rapidminer and the recommendation system on the website. Then it can be concluded that the system can group films with high similarity based on rating, popularity, and profit criteria. Users can select one or more genres as well as the desired year

range. The recommendation system can also be used through various devices because it is web-based so this system can be easily used anywhere.

Keywords: *recommendation system, film recommendation system, k-means clustering method, web.*

1. PENDAHULUAN

Film menjadi satu dari sekian banyak media *entertainment* atau hiburan di zaman ini. Jika kita lihat data pelanggan salah satu platform *online streaming* yang paling populer yaitu *netflix*. Laporan statistika di kuartal III 2021, melaporkan jumlah *user* atau pengguna platform *streaming netflix* terdapat sebanyak 213,56 juta *user* dari setiap negara. Selain *netflix* sekarang ini sudah banyak pilihan platform untuk menonton film mulai dari *hbo*, *disney+*, *hulu*, untuk film-film internasional maupun untuk film-film indonesia atau film lokal di antaranya ada Vidio, WeTV, HOOQ, iflix, dan lain-lain. Film diartikan sebagai suatu genre seni bercerita berbasis audio-visual, atau cerita yang dituturkan pada penonton melalui rangkaian gambar bergerak [1].

Pada umumnya *website* atau platform teknologi yang umumnya dipergunakan dalam mencari film antara lain adalah *google*, *alexa*, *siri* dan sejenisnya. Tetapi masalahnya *website* dan media platform ini hanya dapat mencari dari satu jenis kriteria film. Misal hanya dapat mencari film dengan *rating* tertinggi, atau mencari film dengan tingkat kepopuleran yang tinggi, atau juga mencari film dengan tahun keluarnya film tersebut. jadi tidak bisa dengan kombinasi lebih dari satu kriteria. Juga setiap orang memiliki preferensi yang berbeda beda dalam mencari film yang diinginkan. Ada yang berdasarkan *rating* yang didapatkan suatu film. Walaupun ada juga yang merasa *rating* film kadang tidak relevan atau berbeda dengan pengalaman menonton yang mereka rasakan, ada juga yang berdasarkan popularitas suatu film. Maka dari permasalahan tersebut diperlukan sistem rekomendasi film yang dapat membantu pengguna dalam mencari film-film dengan kombinasi lebih dari satu kriteria film. Juga membantu pengguna dalam mencari preferensi genre yang pengguna inginkan dan mengelompokkan film-film tersebut.

Pada penelitian yang sama oleh Herdita Mutiasari, Tito Waluyo Purboyo, Ratna Astuti Nugrahaeni dengan judul “Sistem Rekomendasi Film Menggunakan Metode *K-Means Clustering*”. Lalu hasil penelitian yang mereka lakukan dapat disimpulkan bahwa aplikasi mampu membantu pengguna untuk menampilkan list film yang sesuai dengan apa yang diinginkan oleh pengguna, analisis *k-means* dapat mempermudah melakukan perekomendasi film sesuai dengan genre dan *rating*, aplikasi menggunakan 4 bobot preferensi pada algoritma *k-means* menghasilkan nilai error yang lebih kecil dengan nilai 0,44% dibandingkan menggunakan 3 atau 5 preferensi, hasil dari genre yang tidak menggunakan bobot hasilnya sama dengan hasil yang menggunakan 3 preferensi genre [2].

Lalu juga pada penelitian dengan tema yang sama oleh Ulfah Indriani dengan Judul “Pendekatan *K-Means Clustering* Terhadap Rekomendasi Film Untuk Ditonton”. Hasil kesimpulan dari penelitian tersebut antara lain adalah *K-means clustering* dapat digunakan untuk mengelompokkan atau mengklasifikasi data tergantung pada tujuannya. Misalnya, mengelompokkan film sehingga kita dapat mengetahui mana yang sangat direkomendasikan untuk ditonton dengan menggunakan data *rating*, *rating* kritikus dan nominasi penghargaan. Film yang direkomendasikan dan tidak disarankan bukan berarti filmnya tidak bagus. Orang bisa menggunakan selera mereka dalam menentukan film mana yang ingin mereka tonton setelah melihat hasil pengelompokan [3].

Dari permasalahan diatas juga penelitian dengan tema yang sama, maka peneliti juga termotivasi untuk melakukan penelitian dengan tema yang sama. penelitian dilakukan dengan tujuan dapat membangun sistem *website* yang dapat membantu pengguna dalam mencari film-film dengan kombinasi lebih dari satu kriteria film. Juga membantu pengguna dalam mencari preferensi genre yang pengguna inginkan dan mengelompokkan film-film tersebut.

2. PEMBAHASAN

2.1. Landasan Teori

HTML ialah sesuatu bahasa yang dikenal dan dipahami oleh web *browser* berguna untuk menyajikan informasi mulai dari teks, suara, gambar, video, bahkan animasi [4]. *Cascading Style Sheet* disingkat CSS merupakan sesuatu Bahasa pemrograman web dimana digunakan untuk mengatur serta membangun bermacam komponen dalam website sehingga tampilan pada web menjadi lebih teratur, terstruktur, serta seragam [5]. *Hypertext Preprocessor* disingkat PHP merupakan sesuatu bahasa pemrograman yang dipakai untuk menerjemahkan basis kode program ke kode mesin yang bisa dipahami oleh komputer yang sifatnya *server-side* dan ditambahkan pada HTML [6]. Javascript merupakan *Scripting language* atau bahasa skrip, ialah kumpulan intruksi perintah, yang dimana intruksi perintah ini digunakan sebagai pengendali dari beberapa bagian pada sistem operasi [7].

Bahasa pemrograman ialah sesuatu sintak atau kumpulan sintak guna mendefinisikan program pada komputer, bahasa pemrograman memungkinkan programmer membuat sesuatu program aplikasi [8]. Web merupakan sistem pada jaringan internet yang memiliki kaitan dengan dokumen, berguna sebagai media yang menampilkan mulai dari teks, foto atau gambar, multimedia, serta yang lainnya [7]. MySQL adalah *Relational Database Management System* (RDBMS) yang dapat memajemen dan mengelola *database* dengan kecepatan tinggi, juga dapat menyimpan data dalam jumlah yang sangat besar, serta dapat dengan mudah diakses oleh banyak pengguna [9]. *RapidMiner* adalah perangkat lunak yang sifatnya terbuka atau open source. perangkat lunak ini merupakan solusi untuk menganalisis data mining, text mining serta analisis prediksi. *RapidMiner* memakai berbagai metode deskriptif serta prediksi dalam membagikan pengetahuan kepada pengguna sehingga bisa membuat keputusan yang sangat baik [10].

Metode yang digunakan pada pembuatan sistem rekomendasi film ini adalah metode *K-Means Clustering*. Metode *k-means* mempartisi data atau objek ke dalam *cluster* atau kelompok. data yang mempunyai ciri yang sama akan dikelompokkan dan dimasukkan ke dalam kelompok atau *cluster* yang sama juga. Ada pula tujuan data *clustering* merupakan untuk meminimalkan *objective function* yang diset atau diatur pada proses *clustering*, yang biasanya berupaya meminimalkan variasi dalam suatu *cluster* serta memaksimalkan variasi dari masing-masing *cluster*. Manfaat dari *clustering* yaitu sebagai Identifikasi Objek (*Recognition*) misal dalam bidang *Image Processing*, *Computer Vision* ataupun *robot vision*[11]. Tidak hanya itu manfaat lain yaitu sebagai SPK (Sistem Pendukung Keputusan) serta *data mining* semacam pemetaan daerah, segmentasi pasar, manajemen marketing dan sebagainya [12]. metode *K-Means* biasanya dilakukan menggunakan algoritma dasar yaitu sebagai berikut:

- Menentukan jumlah *k cluster* atau kelompok.
- Menentukan nilai *centroid* awal dengan mengambil nilai secara acak dari *dataset*.
- Kelompokkan data sebanyak *k* kelompok, kelompokkan dengan menghitung *distance* atau jarak dari masing-masing data ke *centroid* terkecil.
- Menghitung nilai *centroid* baru dengan menghitung rata-rata nilai dari masing-masing kelompok.
- Kembali ke langkah c, jika masih ada data yang berpindah *cluster* atau apabila ada perubahan dari nilai *centroid*.

Menghitung jarak terkecil dari masing-masing data ke *centroid* menggunakan rumus *euclidean distance* yaitu:

$$D_{L_2}(x_2, x_1) = \|x_2 - x_1\|_2 = \sqrt{\sum_{j=1}^p (x_{2j} - x_{1j})^2} \quad (1)$$

Dimana:

D_{L_2} = jarak kuadrat *Euclidean* antar objek ke x_2 dengan x_1 .

P = jumlah variabel *cluster*

X_{2j} = nilai atau data dari objek ke-2 pada variabel ke- j

X_{1j} = nilai atau data dari objek ke-1 pada variabel ke- j .

2.2. Perancangan Sistem Website

2.2.1. Sistem Rekomendasi

```
program rekomendasi

var api_key : string
var base_url : string
var searchUrl : string
var data_sudah_ditonton : string
var genre : string
var tglawal : string
var tglakhir : string
var data : array
var result : array

import kmeans from "./module/kmean.js";
api_key = "api_key=1cf50e6248dc270629e802686245c2c8"
base_url = "https://api.themoviedb.org/3"
searchURL = BASE_URL + "/search/movie?" + API_KEY

INPUT genre
INPUT tglawal
INPUT tglakhir
getMovies(BASE_URL +
"/discover/movie?" +
API_KEY +
"&primary_release_date.gte=" +
tglawal +
"&primary_release_date.lte=" +
tglakhir +
"&with_genres=" +
encodeURIComponent(genre.join(", ")))
)
Function getMovies(url){
fetch(url)
.then((res) => res.json())
.then((data) => {
    id_user = $_SESSION['id_user']
    sql = "SELECT data_film FROM film_ditonton WHERE id='id_user'"
    result = mysqli_query(DATABASE, sql);
    row = mysqli_fetch_assoc(result)
    idfilmditonton= row['data_film']
    data.remove(idfilmditonton)

    keuntungan = data.revenue/data.budget
    data = [data.rating, data.popularity, keuntungan]
    result = kmeans(data, 3)
    showMovies(result)
})
Function showMovies(datafilm){
datafilm.forEach( (movie) =>{
    OUTPUT movie.title
    OUTPUT movie.poster
    OUTPUT movie.rating
    OUTPUT movie.overview
    OUTPUT movie.videos}})
```

Pada program rekomendasi diatas pengguna memasukkan input genre yang diinginkan, juga tanggal awal dan akhir untuk menjadi rentang tanggal film. Inputan tadi akan dikombinasikan dengan variabel-variabel sebelumnya yang berisi *link* untuk berkomunikasi dengan TMDB API. Lalu *link* tadi akan di oper ke fungsi *getMovies()* untuk di *fetch* datanya.

Pada fungsi *getMovies()*, isi dari *url* yang dikirim lewat parameter fungsi akan menjadi *link url* untuk *request* data film ke TMDB API. *url* akan di *fetch* lewat fungsi *javascript*, lalu hasil *fetch* akan di pecah atau dibuat dalam format *json* dengan fungsi *json javascript*. Sekarang format data sudah menjadi kumpulan objek pada array (formatnya berupa `[[],[],[]]`). Ambil data film yang sudah ditonton dari tabel *film_ditonton*. Lalu hapus data film yang sudah ditonton dari hasil *json*. Dengan begitu data film yang sudah ditonton tidak akan muncul lagi pada halaman rekomendasi. Agak berbeda dengan proses pencarian, sebelum data hasil *json* dioper ke fungsi *showMovies*, ada beberapa perintah yang akan dilakukan. Pertama menghitung keuntungan dengan membagi nilai properti *revenue* dengan *budget*. Setelah itu panggil fungsi *kmeans* yang terdapat pada file yang berbeda dan sudah diimport ke program rekomendasi ini. Data yang akan dihitung pada fungsi *kmeans* ada tiga, pertama *rating*, kedua *popularity*, dan ketiga keuntungan yang sudah dihitung sebelumnya. Parameter pada fungsi *kmeans* ini ada dua yang pertama data yang akan dihitung dan kedua jumlah *cluster* atau kelompok. Pada penelitian ini saya menggunakan tiga kelompok. Lalu hasil dari fungsi *kmeans* ini akan dioper ke fungsi *showMovies*.

Fungsi *showMovies()* digunakan untuk menampilkan atau mencetak film-film ke layar. Data film yang di kirim lewat parameter “*datafilm*”, akan Menggunakan *method forEach* pada *datafilm* yang di oper lewat parameter fungsi *showmovies* untuk mencetak setiap film ke layar. Yang dicetak ke layar untuk setiap film antara lain poster, judul, *rating*, *overview* atau sinopsis singkat, dan video trailer

2.2.2. K-Means Clustering

```

Program kmeans
var data : array
var k : integer
var centroid : array
var jumlah : real
var distance : array
var a : integer
var b : integer
var c : integer
var d : integer
var e : integer
var isi : array

INPUT data
INPUT k
IF (data.length && data[0].length > k) THEN
  FOR (var i = 0; i < k ; i++) {
    centroid[i] = data[randomBetween(0, data.length)]
    WHILE (loop) {
      //hitung jarak dengan euclidean distance
      FOR (a = 0; j < k ; j++) {
        FOR (b = 0; i < data.length ; i++) {
          FOR (c = 0; j < k ; j++) {
            Jumlah+=(data[b][c]-centroid[a][c])*
              (data[b][c] - centroid[a][c])
            IF (c == k-1) THEN
              distance[b][a] = Math.sqrt(jumlah)
              Jumlah = 0 }
          }
        }
      }
    }
  }
  //Kelompokkan data dengan Jarak distance ke centroid terkecil

```

```
FOR (a = 0; a < distance.length ; a++){
    cluster[indexOf(distance[0].min)].push(data[a])
//hitung nilai centroid baru
FOR (b = 0; b < cluster.length ; b++){
    FOR (c = 0 ; c < cluster[b].length ; c++){
        FOR (d = 0 ; d < cluster[b][c].length ; d++){
            FOR (e = 0 ; e < cluster[b].length ; e++){
                mean += cluster[b][e][d]
                IF (e == cluster[b].length-1) THEN
                    mean = mean / cluster[b].length
                    centroidBaru[b].push(mean) }}}}
//bandingkan centroid dengan centroid baru
FOR (a = 0 ; a < centroidBaru.length ; a++){
    IF (centroid[a] == centroidBaru[a]) THEN
        isi.push("sama")
    ELSE
        isi.push("beda")}
IF (isi.include("beda")) THEN
    centroid = centroidBaru
    distance = []
    cluster = []
    jumlah = 0
    mean = 0
    centroidBaru = []
ELSE
    loop = false
    OUTPUT cluster
}

ELSE
    OUTPUT "jumlah cluster tidak boleh lebih dari panjang data!"
```

Pada program *kmeans* terdapat dua input yang dibutuhkan. Yang pertama input data yang berupa *array*, dan yang kedua input *k* yaitu banyaknya jumlah *cluster* atau kelompok yang diinginkan. Jumlah *cluster* harus kurang dari panjang data, jika jumlah *cluster* lebih besar dari panjang data maka tampilkan pesan "jumlah *cluster* tidak boleh lebih dari panjang data!". Selanjutnya jika jumlah *cluster* tidak lebih dari panjang data maka lakukan langkah-langkah berikut:

- a. Ambil nilai awal titik pusat data atau *centroid* dari data secara acak.
- b. Lakukan pengulangan.
- c. Hitung jarak antara data dengan *centroid* dengan *euclidean distance*
- d. Kelompokkan data sebanyak *k* kelompok, kelompokkan dengan jarak hasil perhitungan *distance* pada langkah sebelumnya ke *centroid*, dan cari nilai terkecil.
- e. Hitung nilai *centroid* baru dengan menghitung rata-rata dari masing-masing kelompok atau *cluster*.
- f. Jika nilai *centroid* berubah teruskan pengulangan, dan ulang langkah c ke f.
- g. Jika nilai *centroid* tidak berubah maka hentikan pengulangan.

Lalu cetak hasil pengelompokkan atau *cluster* data.

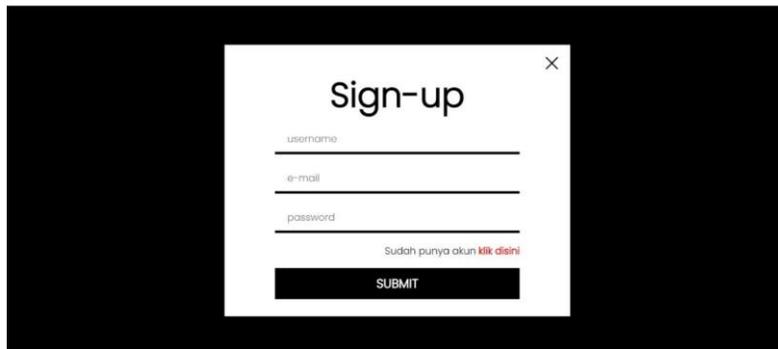
2.3. Tampilan Website

Berikut adalah tampilan-tampilan *website* yang telah dibuat:



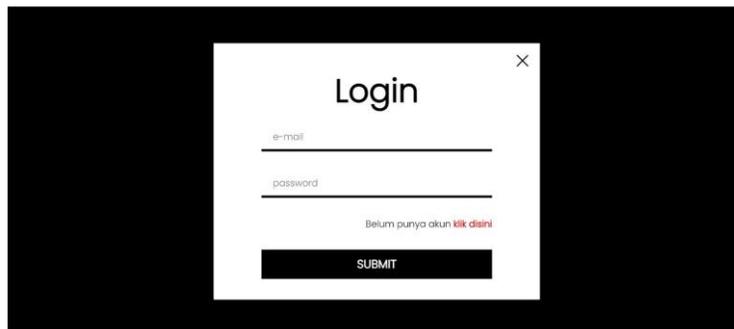
Gambar 1. Tampilan Halaman Awal

Tampilan ini merupakan tampilan halaman awal dari *website* rekomendasi film yang dibangun. pada halaman ini terdapat empat tombol antara lain tombol *home* untuk ke *section home*. Tombol *about* untuk ke *section about*. Tombol *login* untuk ke halaman *login*. Tombol *signup* untuk ke halaman *signup*.



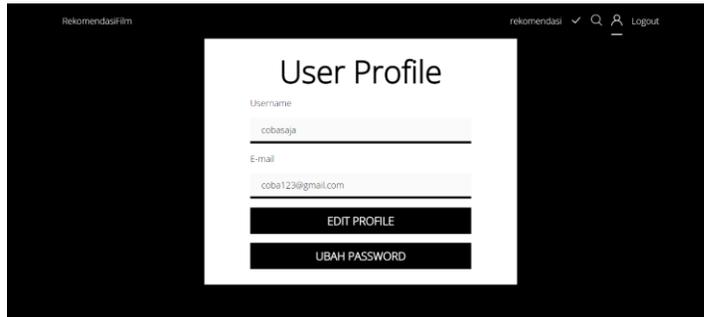
Gambar 2. Tampilan Halaman Signup

Pada halaman *signup* ini berfungsi untuk mendaftarkan akun pengguna ke sistem *website*.



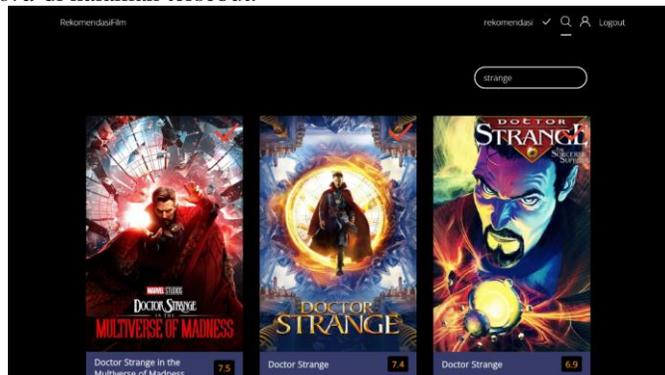
Gambar 3. Tampilan Halaman login

Pada halaman *login* berfungsi untuk melakukan *login* untuk masuk ke *website* rekomendasi film.



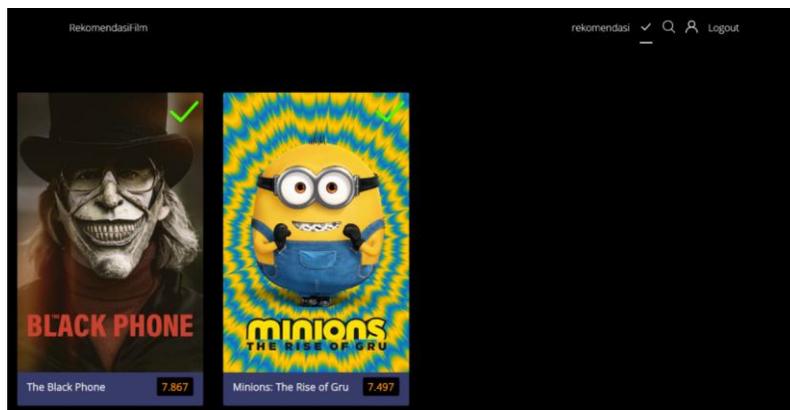
Gambar 4. Tampilan Halaman *profile*

Pada halaman *profile*, pengguna dapat melihat informasi akun saat ini. Juga dapat mengubah informasi akun dengan mengklik tombol *edit profile* untuk masuk ke halaman *edit profile* dan mengganti informasi akun pada halaman tersebut. Lalu pengguna juga dapat mengubah *password* dengan menekan tombol *ubah password* untuk masuk ke halaman *ubah password* dan mengganti *password* di halaman tersebut.



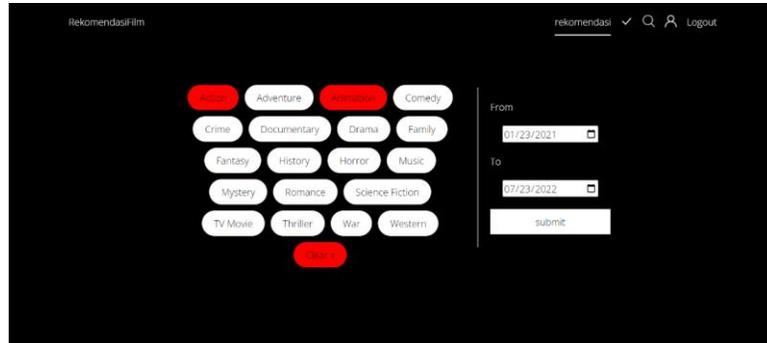
Gambar 5. Tampilan Halaman Pencarian

Pada halaman ini pengguna dapat mencari film yang dicari berdasarkan *keyword* atau nama film. Contohnya dapat dilihat pada gambar diatas.



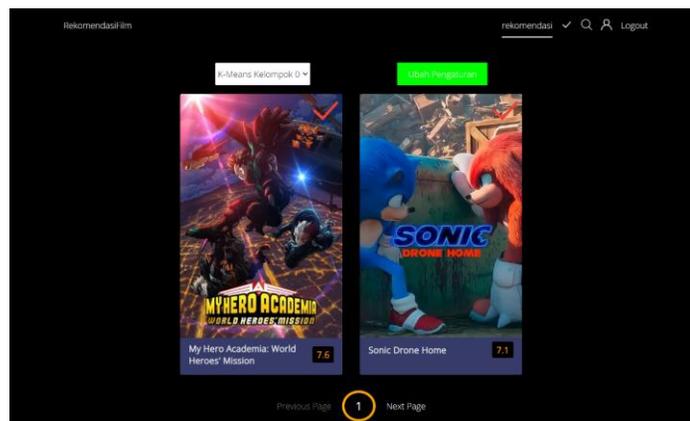
Gambar 6. Tampilan Halaman Film Ditonton

Pada halaman film ditonton terdapat informasi film yang sudah ditonton oleh pengguna. cara menambahkan film ke halaman film ditonton dapat dilakukan dengan mengklik ikon ceklis berwarna merah yang terdapat di halaman pencarian juga di halaman rekomendasi. Dan untuk menghapus film dari halaman film ditonton dengan mengklik ikon hijau yang terdapat pada poster.



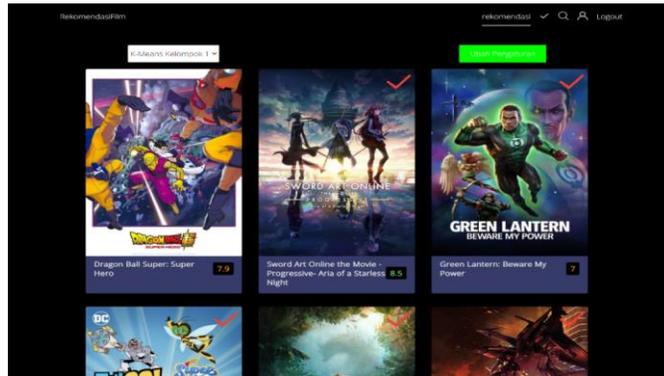
Gambar 7. Tampilan Halaman Rekomendasi Menu

Berikut ini adalah tampilan awal pada halaman rekomendasi. Dapat dilihat pada gambar diatas terdapat menu pengaturan untuk meminta rekomendasi film sesuai dengan genre dan tahun awal juga tahun akhir.



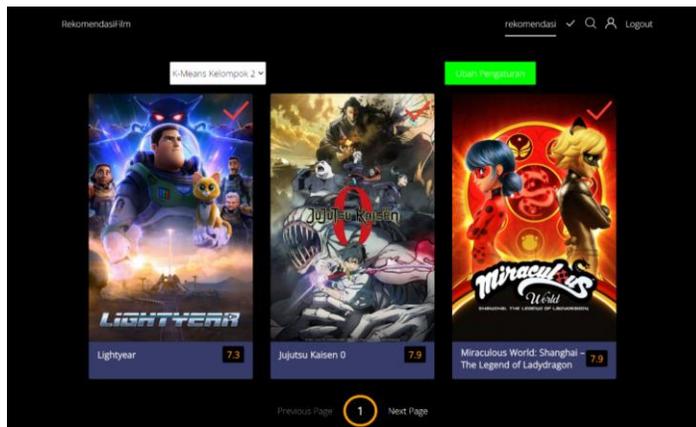
Gambar 8. Tampilan Halaman Rekomendasi Hasil Pengelompokkan Kelompok 0

Jika menu pengaturan sudah diatur sesuai keinginan pengguna lalu tombol *submit* diklik maka akan muncul tampilan seperti pada gambar diatas dapat dilihat hasil kelompok 0 dari pengelompokkan dengan *k-means clustering*. terdapat tombol *select* untuk menentukan kelompok mana yang ingin dilihat mulai dari kelompok 0, 1 dan 2. Lalu disamping tombol *select* terdapat tombol ubah pengaturan berwarna hijau yang jika ditekan akan mengembalikan halaman ke halaman pengaturan awal, untuk menentukan genre juga rentang tahun.



Gambar 9. Tampilan Halaman Rekomendasi Hasil Pengelompokkan Kelompok 1

Tampilan seperti pada gambar diatas dapat dilihat hasil kelompok 1 dari pengelompokkan dengan *k-means clustering*. terdapat tombol *select* untuk menentukan kelompok mana yang ingin dilihat mulai dari kelompok 0, 1 dan 2. Lalu disamping tombol *select* terdapat tombol ubah pengaturan berwarna hijau yang jika ditekan akan mengembalikan halaman ke halaman pengaturan awal, untuk menentukan genre juga rentang tahun.



Gambar 10. Tampilan Halaman Rekomendasi Hasil Pengelompokkan Kelompok 2

Tampilan seperti pada gambar diatas dapat dilihat hasil kelompok 0 dari pengelompokkan dengan *k-means clustering*. terdapat tombol *select* untuk menentukan kelompok mana yang ingin dilihat mulai dari kelompok 0, 1 dan 2. Lalu disamping tombol *select* terdapat tombol ubah pengaturan berwarna hijau yang jika ditekan akan mengembalikan halaman ke halaman pengaturan awal, untuk menentukan genre juga rentang tahun.

2.4. Uji Coba Data Pada Website

Pada uji coba ini akan dilakukan dua uji coba. Pertama uji coba manual dengan bantuan *software RapidMiner*. Dan uji coba program pada *website*. Data yang digunakan pada dua ujicoba ini berasal dari *response* TMDB API dengan genre *action* dan *comedy*, dan tanggal dari 01/01/2015 hingga 01/01/2022. Menggunakan *link* berikut untuk meminta data film ke TMDB API:

https://api.themoviedb.org/3/discover/movie?api_key=81c9ca6f6006c9ae974edcf29552cfe5&primary_release_date.gte=2015-01-01&primary_release_date.lte=2022-01-01&with_genres=28,35

Tulisan atau *link* yang berwarna ungu adalah *api key* yang digunakan untuk meminta data ke TMDB API *api key* ini didapatkan setelah melakukan daftar akun dan juga *request api key* pada *website* TMDB (themoviedb.org). tulisan yang berwarna merah adalah tanggal awal juga tanggal akhir dari film diisikan sesuai dengan data yang akan diujikan yaitu dari tanggal 01/01/2015 hingga 01/01/2022. Sedangkan yang berwarna biru adalah *id* genre, dimana 28 adalah *id* genre *action* dan 35 adalah *id* genre *comedy*. Setelah link diatas dimasukkan pada browser lalu klik enter maka akan menampilkan data-data film hasil respon dari TMDB API.

Lalu data yang akan diambil dari masing-masing film dari data hasil respon TMDB API adalah nama film, *vote* atau *rating*, popularitas dan keuntungan yaitu *revenue* dibagi *budget*. Data-data ini dipindahkan ke *microsoft excel* secara manual untuk nantinya diuji menggunakan *rapidminer*. Data yang sudah dimasukkan ke dalam *excel* dapat dilihat sebagai berikut:

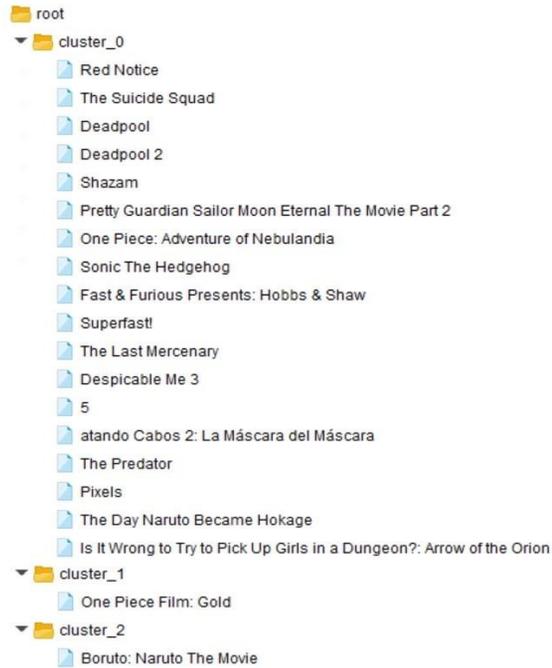
film	Rating	Popularitas	Keuntungan
Red Notice	6.774	306.04	0.001113394
The Suicide Squad	7.6	287.074	0.904959054
Boruto: Naruto The Movie	7.6	243.712	40200000
Deadpool	7.601	221.23	13.50172414
Deadpool 2	7.499	187.982	7.148778527
Shazam	7.1	171.315	4.576000613
Pretty Guardian Sailor Moon Eternal The Movie Part 2	8.1	162.34	0
One Piece: Adventure of Nebulandia	7.6	157.257	0
Sonic The Hedgehog	7.4	142.329	3.609017294
Fast & Furious Presents: Hobbs & Shaw	6.878	118.667	3.80049498
Superfast!	5	100.397	0
The Last Mercenary	6.6	94.867	0
Despicable Me 3	6.4	102.144	12.89440731
5	6.2	101.466	0
atando Cabos 2: La Máscara del Máscara	7.7	103.305	0
One Piece Film: Gold	7.3	94.31	66207073
The Predator	5.6	89.101	1.824342432
Pixels	5.7	75.105	2.782668284
The Day Naruto Became Hokage	7.6	88.255	0
Is It Wrong to Try to Pick Up Girls in a Dungeon?: Arrow of the Orion	7.8	86.791	0

Gambar 11. Data Yang Akan Diujikan

Data diatas adalah data yang akan digunakan pada dua ujicoba berasal dari *response* TMDB API dengan genre *action* dan *comedy*, dan tanggal dari 01/01/2015 hingga 01/01/2022 yang sudah dipindahkan secara manual ke *microsoft excel* jumlah data berupa 20 data film.

2.4.1. Uji Coba Dengan *Rapidminer*

Uji coba dengan *RapidMiner* dengan data yang sudah berupa *excel*. Dengan setting *clustering* sebanyak 3 *cluster*. hasilnya sebagai berikut:



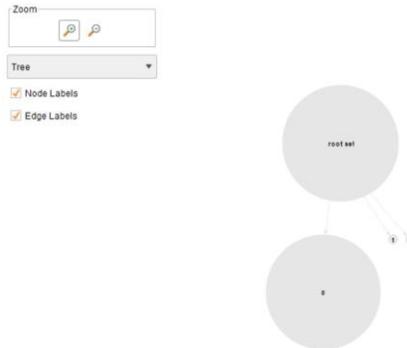
Gambar 12. Root Folder Hasil Kmeans Clustering Dengan Rapidminer

Gambar diatas adalah hasil dari pengelompokkan *k-means clustering* menggunakan *rapidminer*. Hasil pengelompokkan pada gambar diatas ditambahkan dengan bentuk folder.

Attribute	cluster_0	cluster_1	cluster_2
Rating	6.953	7.300	7.600
Popularitas	144.204	94.310	243.712
Keuntungan	2.836	66207073	40200000

Gambar 13. Centroid Table Hasil Kmeans Clustering Dengan Rapidminer

Gambar diatas adalah nilai *centroid* akhir dari pengelompokkan *k-means clustering* menggunakan *rapidminer*. Nilai *centroid* ditampilkan pada gambar diatas ditampilkan dengan berbentuk tabel.

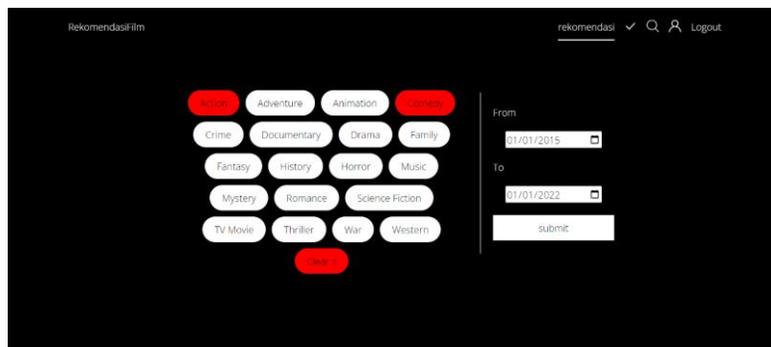


Gambar 14. Graph Hasil Kmeans Clustering Dengan Rapidminer

Gambar diatas adalah hasil dari pengelompokkan *k-means clustering* menggunakan *rapidminer*. Hasil pengelompokkan pada gambar diatas ditampilkan dalam bentuk *graph*.

2.4.2. Uji Coba Pada Website

Sedangkan pada uji coba pada *website* tidak perlu mengatur *link API* juga memindahkan data-data film ke *excel* secara manual seperti pada pengujian *rapidminer*. Pada pengujian *website* hanya tinggal menekan tombol genre *action* dan *comedy* dan juga mengatur tanggal awal (*form*) yaitu 01/01/2015 dan tanggal akhir (*to*) 01/01/2022. Proses bagaimana sistem mengolah hasil input ini juga bagaimana sistem mengelompokkan data hasil *response* TMDB API tersebut menggunakan algoritma *k-means clustering* dapat dilihat pada bagian sebelumnya pada jurnal ini yaitu *pseudocode* sistem rekomendasi dan juga *pseudocode k-means clustering* serta pembahasan algoritma dari masing-masing algoritma *pseudocode*-nya di bagian sebelumnya dari jurnal ini.



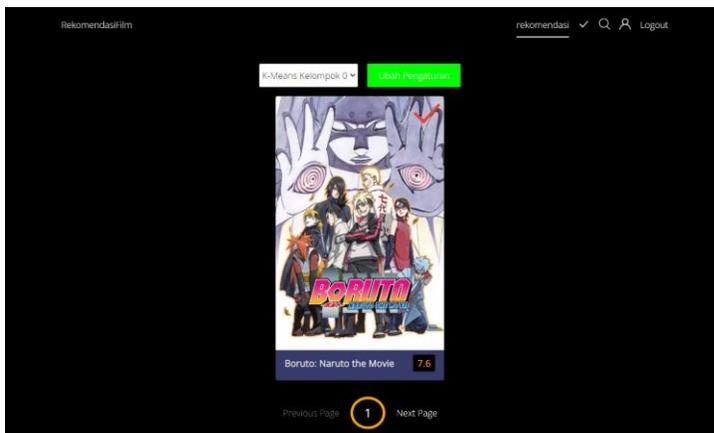
Gambar 15. Setting Genre dan Tanggal Sesuai Dengan Pengujian Dengan Rapidminer

Gambar diatas merupakan tampilan menu halaman rekomendasi dari *website*. Pada pengujian ini dipilih genre *action* dan *comedy* lalu tanggal *form* dimasukkan tanggal 01/01/2015 dan tanggal *to* dimasukkan 01/01/2015, sama dengan pengujian sebelumnya menggunakan *rapidminer*.

```
▼ Object halaman_rekomendasi.js:275  
  ▼ centroids: Array(3)  
    ▶ 0: (3) [7.6, 243.712, 40200000]  
    ▶ 1: (3) [6.952888888888888, 144.20361111111117, 2.8357503348908084]  
    ▶ 2: (3) [7.3, 94.31, 66207073]  
      length: 3  
    ▶ [[Prototype]]: Array(0)  
  ▼ clusters: Array(3)  
    ▶ 0: {points: Array(1), centroid: Array(3)}  
    ▶ 1: {points: Array(18), centroid: Array(3)}  
    ▶ 2: {points: Array(1), centroid: Array(3)}  
      length: 3  
    ▶ [[Prototype]]: Array(0)  
    converged: true  
    iterations: 3  
    ▶ [[Prototype]]: Object
```

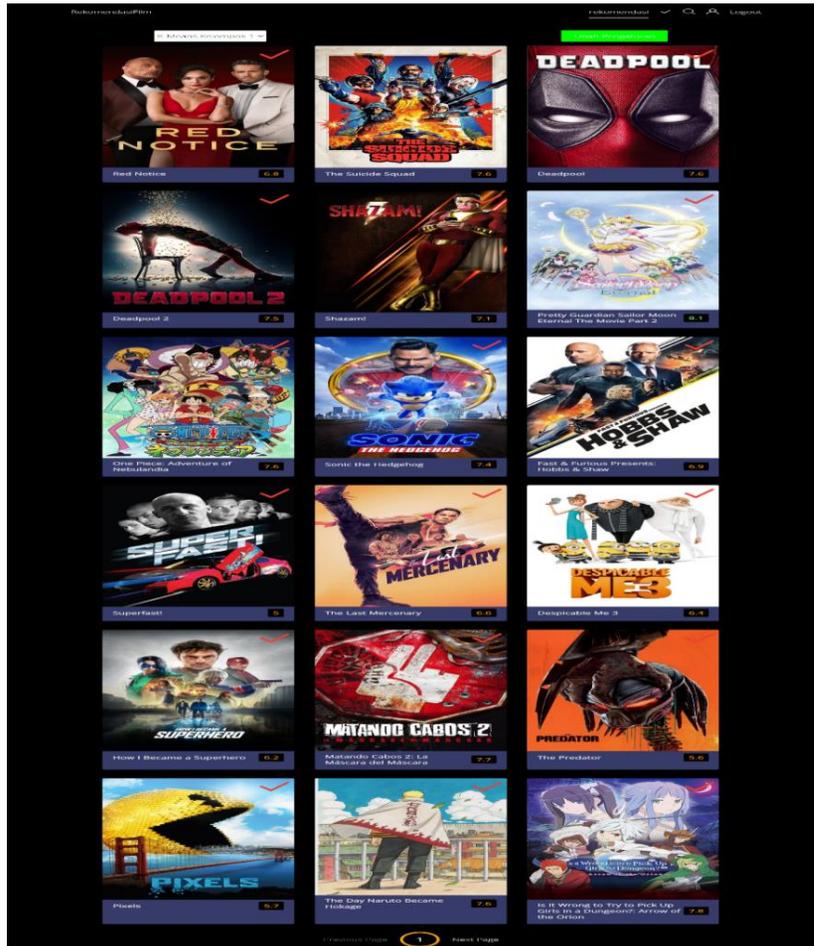
Gambar 16. Screenshot Pada Console Website Dapat Dilihat Centroid Table Hasil Pengujian Dengan Website

Gambar diatas adalah hasil *screenshot console browser*, untuk melihat nilai akhir *centroid* dari pengelompokkan dengan website ini.



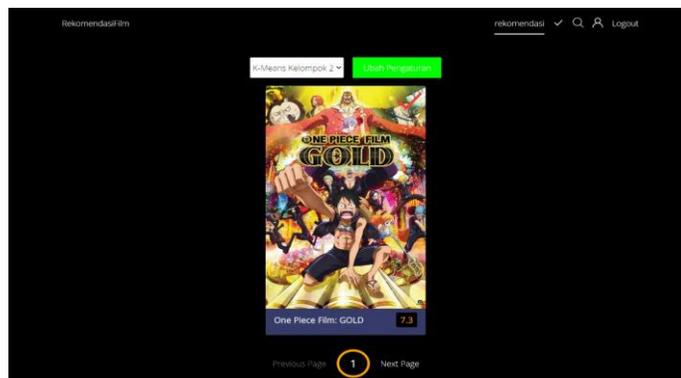
Gambar 17. Kelompok 0 Hasil Pengujian Dengan Website

Gambar diatas dapat dilihat hasil kelompok 0 dari pengelompokkan dengan *k-means clustering* dengan *website*. terdapat tombol *select* untuk menentukan kelompok mana yang ingin dilihat mulai dari kelompok 0, 1 dan 2. Lalu disamping tombol *select* terdapat tombol ubah pengaturan berwarna hijau yang jika ditekan akan mengembalikan halaman ke halaman pengaturan awal, untuk menentukan genre juga rentang tahun.



Gambar 18. Kelompok 1 Hasil Pengujian Dengan Website

Gambar diatas dapat dilihat hasil kelompok 1 dari pengelompokan dengan *k-means clustering* dengan *website*. terdapat tombol *select* untuk menentukan kelompok mana yang ingin dilihat mulai dari kelompok 0, 1 dan 2. Lalu disamping tombol *select* terdapat tombol ubah pengaturan berwarna hijau yang jika ditekan akan mengembalikan halaman ke halaman pengaturan awal, untuk menentukan genre juga rentang tahun.



Gambar 19. Kelompok 2 Hasil Pengujian Dengan Website

Gambar diatas dapat dilihat hasil kelompok 2 dari pengelompokkan dengan *k-means clustering* dengan *website*. terdapat tombol *select* untuk menentukan kelompok mana yang ingin dilihat mulai dari kelompok 0, 1 dan 2. Lalu disamping tombol *select* terdapat tombol ubah pengaturan yang jika ditekan akan mengembalikan tampilan awal halaman, untuk menentukan genre juga rentang tahun.

2.5. Pembahasan Hasil Uji Coba

Dari hasil dua uji coba diatas yaitu uji coba dengan *rapidminer* dan uji coba dengan *website* dapat dilihat bahwa hasil kelompok dari pengelompokkan *k-means clustering* pada *website* memiliki hasil yang sama dengan pengujian data dengan menggunakan perangkat lunak *RapidMiner*. Dapat dilihat pada gambar 12, 17, 18 dan 19 bahwa isi dari masing-masing kelompok film dari hasil pengelompokkan dengan *rapidminer* memiliki isi yang sama dengan kelompok dengan pengelompokkan dengan *website* hanya sedikit berbeda urutan kelompoknya saja. Lalu untuk nilai *centroid*, dapat dilihat dari gambar 13 dan gambar 16 bahwa nilai akhir *centroid* memiliki nilai yang sama antara pengujian dengan *RapidMiner* dan pengujian menggunakan *website*.

3. KESIMPULAN

Berdasarkan hasil perancangan, pembahasan dan pengujian sistem yang dilakukan. Maka dapat disimpulkan sebagai berikut:

- Sistem rekomendasi film dengan metode *k-means clustering* berbasis web berhasil dibuat.
- Sistem dapat mengelompokkan film-film dengan kemiripan tinggi berdasarkan kriteria *rating*, popularitas, dan keuntungan.
- Pengguna dapat memilih satu atau lebih genre, dan rentang tahun yang diinginkan.
- Sistem rekomendasi dapat digunakan lewat berbagai *device* karena berbasis web. Juga membuat sistem ini mudah digunakan dimana saja.

DAFTAR PUSTAKA

- [1] M. I. Zoebazary, *Kamus Televisi & Film*. Jawa Timur: Paguyupan Pandhalungan Jember, 2016.
- [2] H. Mutiasari, T. W. Purboyo, and R. A. Nugrahaeni, "Sistem Rekomendasi Film Menggunakan Metode K-Means Clustering," *e-Proceeding Eng.*, vol. 8, no. 5, pp. 6755–6764, 2021.
- [3] U. Indriani, S. Informasi, U. Potensi, and U. Medan, "Pendekatan K-Means Clustering Terhadap," vol. 2, no. 1, pp. 84–88, 2018.
- [4] Y. K. Ardhana, *PHP Menyelesaikan Website 30 Juta*. Jakarta: Jasakom, 2012.
- [5] T. Wahyudi, "Rancang Bangun Sistem Informasi Pondok Pesantren (Studi Kasus Darul Abror Watumas)," *Simp. Nas. Ilmu Pengetah. dan Teknol. 2017*, vol. 1, no. 1, pp. 23–30, 2017.
- [6] Supono and V. Putratama, *Pemrograman WEB dengan menggunakan PHP dan framework codeigniter*. Yogyakarta: Deepublish, 2016.
- [7] A. F. . Sibero, *Web Programing Power Pack*. Yogyakarta: mediaKom, 2013.
- [8] S. Dipraja, *Panduan Praktis Membuat Website Gratis*. Jakarta: Pustaka Makmur, 2014.
- [9] B. Raharjo, *Belajar Otodidak MySql*. Bandung: Informatika, 2015.

- [10] D. A. C, D. A. Baskoro, L. Ambarwati, and I. W. S. Wicaksana, *Belajar Data Mining dengan RapidMiner*. 2013.
- [11] A. Kroner, M. Senden, K. Driessens, and R. Goebel, “Contextual encoder–decoder network for visual saliency prediction,” *Neural Networks*, vol. 129, pp. 261–270, 2020, doi: 10.1016/j.neunet.2020.05.004.
- [12] U. Khalsum, “Implementasi Sistem Pengelompokan Data Penerima Beasiswa Menggunakan Metode K-MEANS,” Universitas Islam Negeri Sultan Syarif Kasim Riau, 2014.