

LAPORAN TUGAS AKHIR
SEMANTIC WEB
“MovieDatabase”



Oleh :
SABRI SANGJAYA 160535611819

FAKULTAS TEKNIK
JURUSAN TEKNIK ELEKTRO
PRODI S1 TEKNIK INFORMATIKA
UNIVERSITAS NEGERI MALANG
NOVEMBER 2019

BAB I

PENDAHULUAN

1.1 Latar Belakang

Katalog film berbasis web atau sering disebut Katalog Elektronik sebagai salah satu sistem informasi memiliki peran yang sangat vital dalam usaha pengumpulan, pengelolaan dan penyebaran informasi kepada pengguna yang membutuhkannya. Informasi adalah hal yang sangat penting dalam berbagai bidang, termasuk dalam informasi mengenai film. Suatu informasi harus mempunyai tiga syarat yang sangat penting yaitu akurat, tepat dan dapat dipercaya. Untuk memenuhi kebutuhan informasi masyarakat tersebut dengan baik perlu dilakukan optimalisasi dengan cara mengoptimalkan pemanfaatan teknologi informasi (teknologi komputer dan telekomunikasi) untuk mendukung operasional kegiatan mengenai Katalog Elektronik. Dari latar belakang tersebut penulis tertarik untuk membuat suatu sistem informasi katalog film online berbasis web yang bertujuan untuk memudahkan pengguna dalam mengakses informasi dan melakukan pencarian koleksi film secara online melalui internet.

1.2 Tujuan

Tujuan pembuatan website sistem informasi ini antara lain untuk memberikan informasi lengkap mengenai film, yang dapat digunakan sebagai ensiklopedia, dan media promosi film. Website ini juga dapat digunakan sebagai media penunjang rekomendasi film, dikarenakan terdapat informasi mengenai review film yang disertakan.

BAB II

KAJIAN PUSTAKA

2.1 MovieDatabase

MovieDatabase adalah sebuah katalog film berbasis web. Katalog berasal dari bahasa Latin *catalogus* yang berarti daftar barang atau benda yang disusun untuk tujuan tertentu. Katalog merupakan daftar dari koleksi perpustakaan yang disusun secara sistematis, sehingga memungkinkan pengguna dapat mengetahui dan menemukan dengan mudah koleksi apa yang dimiliki oleh perpustakaan

2.2 Ontologi (OWL)

OWL merupakan Bahasa yang telah direkomendasikan oleh W3C merupakan Bahasa ontologi yang paling ekspresif dan digunakan pada aplikasi semantic web untuk merepresentasikan sendiri dari suatu terminology-terminology dalam domain dalam pengetahuan secara eksplisit serta relasi antar terminology tersebut yang tidak mungkin terpisahkan dengan dokumen dan aplikasi web tersebut. Karena web sendiri memiliki visi yaitu memberukan informasi yang bermakna dan eksplisit dan mesin dapat memproses informasi secara otomatis dan mengintegrasikan informasi tersebut pada web. Maka diperlukan Bahasa yang dapat merepresentasikannya dengan menggunakan OWL.

OWL sendiri digunakan untuk merepresentasikan kosakata dan relasi antar kata sehingga menjadi eksplisit. Dan OWL ini dibangun dari RDF dan RDFS serta dengan menambahkan vocabulary agar lebih mendefinisikan classes, property dan individual.

2.3 RDF (Resource Description Framework)

RDF atau Resource Description Framework merupakan bagian dari W3C dengan tujuan untuk model metadata. Konsep dari RDF sendiri memiliki kesamaan dengan konsep sebelumnya. RDF sendiri merupakan model acak dengan menggunakan file format dan banyak variasi kodingnya. Teori ini membuat model data dari RDF lebih baik dari macam-macam representasi dari relasional maupun dari model ontologinya.

```

<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:si="https://www.w3schools.com/rdf/">
  <rdf:Description rdf:about="https://www.w3schools.com">
    <si:title>W3Schools</si:title>
    <si:author>Jan Egil Refsnes</si:author>
  </rdf:Description>
</rdf:RDF>

```

Tabel 1.1 Contoh sintaks dari RDF

Grafik RDF berguna untuk merepresentasikan dari berbagai macam koleksi dari berbagai macam dari statemen. Grafik ini untuk merepresentasikan sebuah relasi diantara entitas

2.4 PHP

PHP merupakan singkatan “PHP Hypertext Preprocessor”, adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML. PHP juga banyak diaplikasikan untuk pembuatan program-program seperti sistem informasi klinik, rumah sakit, akademik, keuangan, manajemen aset, manajemen bengkel dan lain-lain. Dapat dikatakan bahwa program aplikasi yang dulunya hanya dapat dikerjakan untuk desktop aplikasi, PHP sudah dapat mengerjakannya. Penerapan PHP saat ini juga banyak ditemukan pada proyek-proyek pemerintah seperti e-budgetting, e-procurement, e-goverment dan e e lainnya.

BAB III

METODE

3.1 Tahapan Pengembangan

Situs web ini dikembangkan dengan menggunakan metodologi RAD (Rapid Application Development), yang merupakan model proses pengembangan perangkat lunak sekuensial linier dalam periode pengembangan yang sangat singkat (sekitar 60 hingga 90 hari). Tahapan proses pengembangan dengan metode RAD adalah sebagai berikut:

a. **Model Bisnis**

Tahapan ini berisi pengamatan tentang informasi yang perlu dikumpulkan untuk mendukung pengembangan situs web yang dibuat.

b. **Pemodelan Data**

Pada tahap ini, penyaringan informasi yang diperoleh menjadi objek data yang dibutuhkan. karakteristik masing-masing objek akan diidentifikasi dan kemudian hubungan antara objek yang dicari.

c. **Pemodelan Proses**

Objek yang telah ditentukan kemudian diubah untuk mencapai arus informasi yang dibutuhkan dalam implementasi situs web.

d. **Pembuatan Aplikasi**

Pada tahap ini proses pembuatan website dimulai, dengan memperhatikan informasi dan obyek yang diperoleh pada tahap sebelumnya.

e. **Pengujian dan Perbaikan**

Pada tahap ini situs web diuji. jika sesuai dengan target yang diinginkan, perangkat lunak dapat divalidasi. tetapi jika masih ada ketidaksesuaian, situs web akan diperbaiki lagi.

3.2 Analisis Kebutuhan

Analisis kebutuhan sistem adalah tahap awal dan utama untuk membuat pondasi dalam langkah-langkah pengembangan sistem. Analisis kebutuhan sistem mencakup dua hal utama yang harus diperhatikan, yaitu analisis kebutuhan fungsional dan analisis kebutuhan nonfungsional.

3.3.1 Kebutuhan Fungsional

Analisis kebutuhan fungsional mengacu pada panduan belajar, literatur, dan observasi langsung. jadi analisis ini dapat dikatakan lebih mengacu pada sistem

yang ada. Dalam hal mengembangkan situs web sistem informasi film, setidaknya ada persyaratan proses, antara lain:

- a. Sebuah sistem dapat menampilkan informasi film berdasarkan data yang telah diunggah ke server / database.
- b. Sistem dapat menampilkan informasi ulasan / komentar dari pengguna lain (jika ada) di setiap post.
- c. Pengguna dapat melakukan pencarian yang akan memberikan hasil dalam bentuk post terkait.

3.3.2 Kebutuhan Non-fungsional

Kebutuhan non-fungsional fokus pada properti yang menyertai berjalannya sistem pada kebutuhan ini, ada beberapa titik referensi seperti operasional, keamanan, informasi, kinerja. Untuk analisis persyaratan nonfungsional di situs web tersebut, antara lain:

a. Operasional

Deskripsi dari nilai minimum yang harus dipenuhi untuk menjalankan situs web ini meliputi sistem operasi Windows 7, RAM 2GB, koneksi internet, browser dengan versi terbaru.

b. Informasi

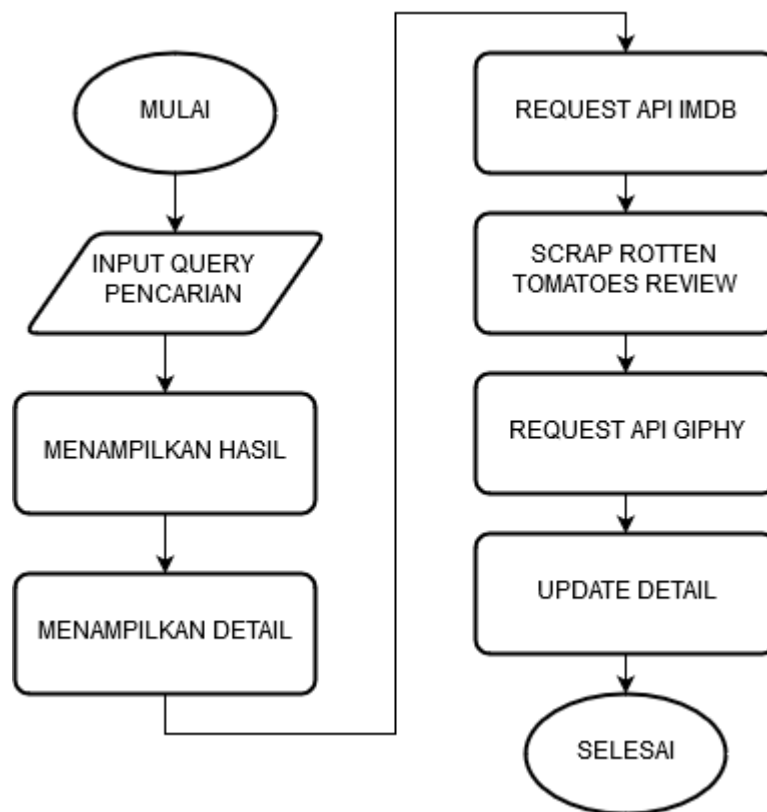
Informasi validasi ketika gagal atau berhasil mengambil data dari database (pencarian).

c. Kinerja

Konten situs web ini akan diperbarui secara berkala, ketika terdapat bug / error sistem dalam website akan diperbaiki secara continuous software development.

3.3 Desain Sistem

3.3.1 Desain Flowchart

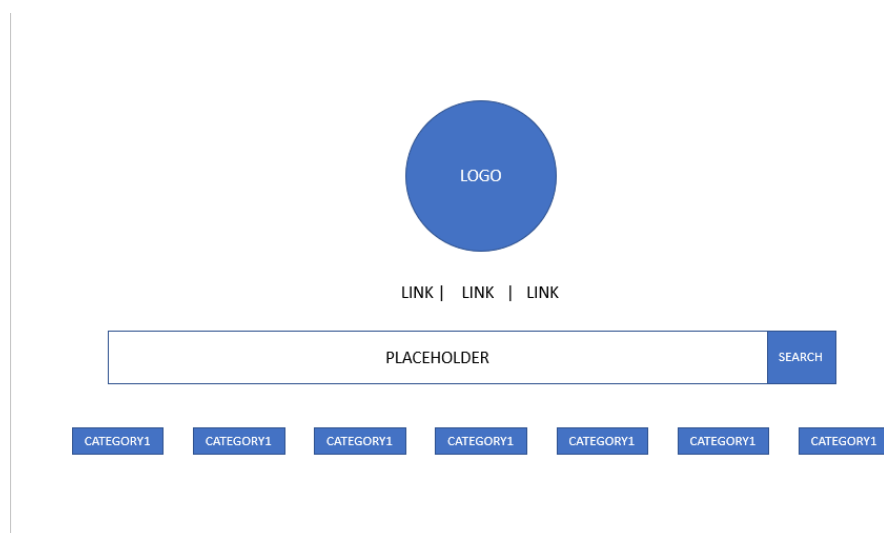


Gambar 3.1 Flowchart

Gambar di atas menunjukkan alur penggunaan situs web yang kami buat. mulai dari proses pencarian data hingga proses penampilan hasil dan detail pencarian.

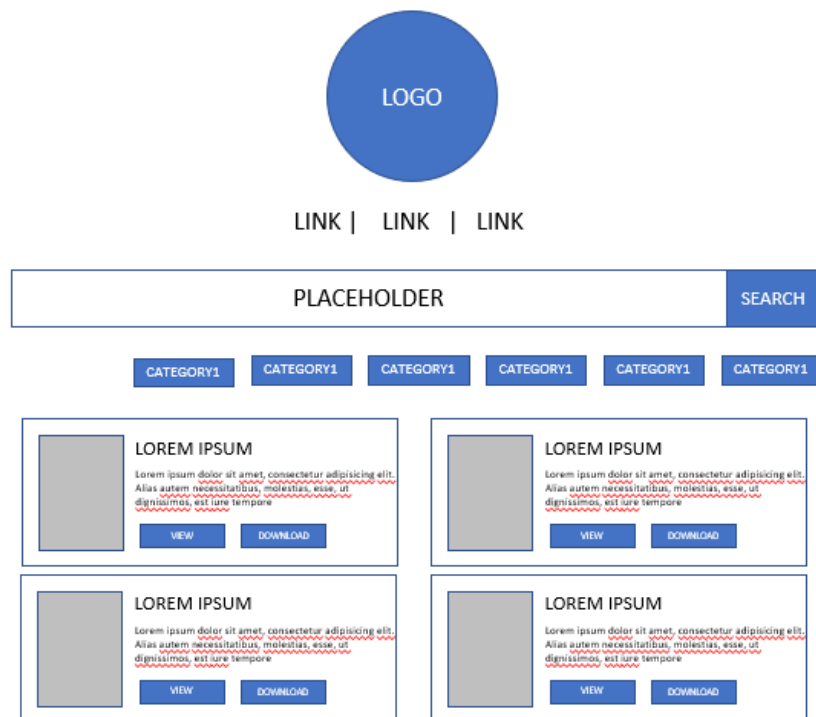
3.3.2 Desain Tampilan

Berikut ini merupakan desain layout yang kemudian akan digunakan pada tahap implementasi. Setiap bagian dari masing-masing halaman akan dijelaskan, agar dapat dimengerti oleh pembaca



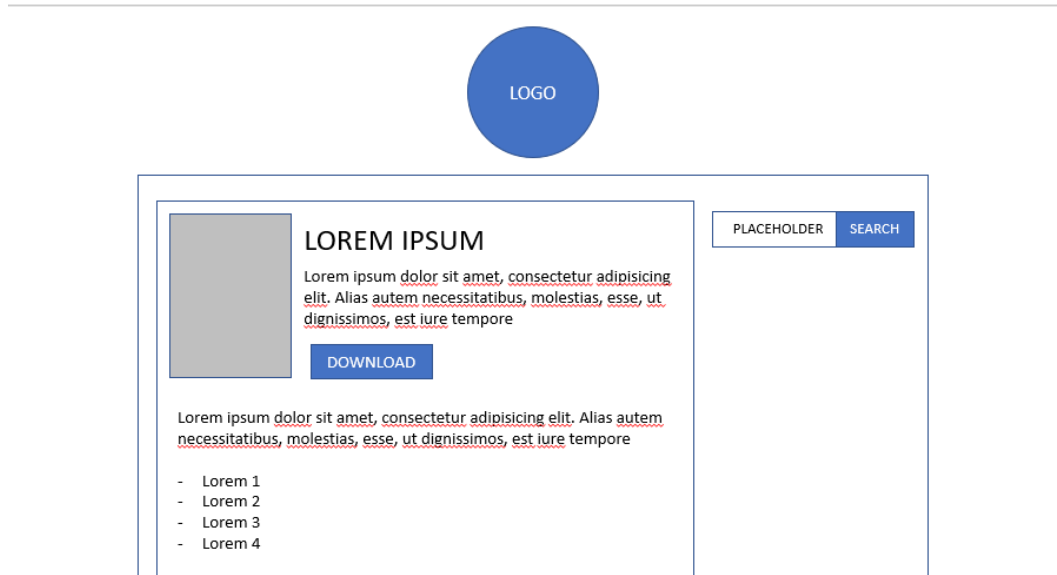
Gambar 3.2 Halaman Awal

Halaman utama website ini terdapat bagian header dan menu navigasi dimana jika dipilih akan mengarahkan pengguna ke halaman sesuai pilihan. Terdapat pula form pencarian bagi pengguna agar lebih mudah menemukan hal yang diinginkan.



Gambar 3.2 Halaman Hasil Pencarian

Tampilan diatas menunjukkan bagian dari halaman katalog atau daftar film dari website ini. Halaman ini juga menampilkan hasil fungsi pencarian, desain dari halaman ini sangat minimalis yang hanya terdiri dari form input keyword pencarian, dan dibagian bawah merupakan daftar film yang ada pada website ini .Halaman ini terdiri dari bagian header, dan bagian konten. Bagian header mirip dengan yang ada pada halaman utama, perbedaannya ada pada ilustrasi yang digunakan. Bagian “view” pada setiap post yang ada pada halaman katalog diatas, jika dipilih akan mengarahkan pengguna pada tampilan penuh satu film yang berkaitan. Berikut dibawah ini merupakan desain dari halaman tersebut.



Gambar 3.2 Halaman Detail Film

BAB IV

HASIL DAN PEMBAHASAN

4.1.Deskripsi Sistem

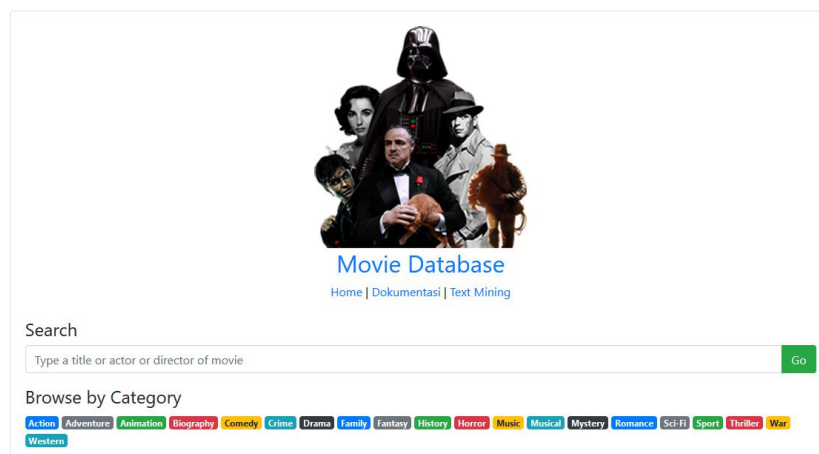
Website ini dibuat dengan menggunakan PHP. Proses impleentasi desain dari website ini, dibantu dengan menggunakan framework Bootstrap untuk memudahkan dan mempercepat proses implementasi. Fitur utama dari situs ini adalah pencarian film berdasarkan kata kunci user. Konsep semantik web dalam penelitian ini dibuat menggunakan OWL (*Ontology Web Language*) yang terdiri dari RDF (*Resource Description Framework*) Setelah proses pembuatan ontologi selesai. Membuat situs web dengan fungsi dan fitur yang direncanakan. Konten yang terdapat di situs web disesuaikan dengan konsep RDF yang telah dibuat (sesuai dengan kelas yang ada). Ontologi yang telah dibuat tersebut kemudian dijadikan sebagai landasan pembuatan susunan data dalam database, serta konten yang terdapat pada website ini.

4.2.Source Code Sistem

Source Code Implementasi dari website ini dilampirkan pada **Lampiran I**.

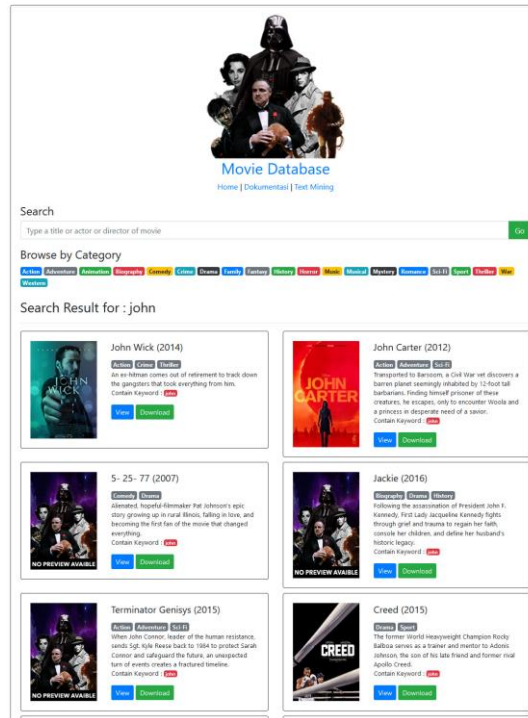
4.3.Interface Sistem

Proses implemenntasi desain UI (User Interface) yang telah dilakukan menghasilkan tampilan yang kurang lebih mirip dengan desain yang dijadikan rujukan, namun masih terdapat beberapa perubahan yang disesuaikan dengan kebutuhan.

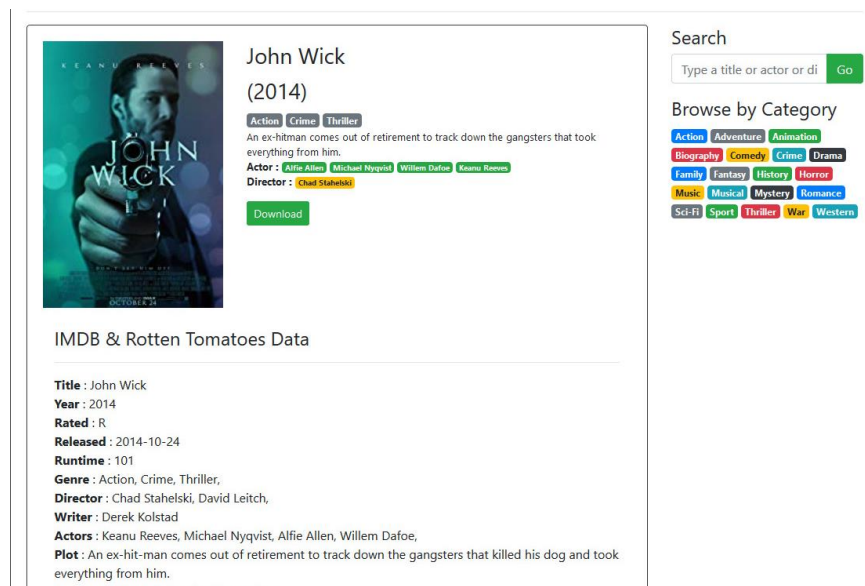


Gambar 4.1 Interface Halaman Awal

Halaman fitur paling penting pada website ini adalah pencarian yang sekaligus halaman awal, halaman ini terdiri dari form input untuk query pencarian.



Gambar 4.2 Interface Halaman Hasil Pencarian



Gambar 4.3 Interface Halaman Detail Film

4.4. Analisis Sistem

Dalam proses pembuatan ontologi yang ada pada website ini, Tahap pertama adalah untuk menentukan domain dan batasan ontologi yang akan dibuat. Kemudian menulis daftar istilah yang akan dijelaskan kepada pengguna. Langkah selanjutnya adalah mendefinisikan kelas dari ontologi, dan mengatur kelas dalam hirarki taksonomi (subclass - superclass). Pengembangan hirarki kelas dalam penelitian ini menggunakan model top-down, yang

dimulai dengan mendefinisikan konsep umum dalam domain dan berlanjut dengan konsep yang lebih spesifik. Langkah terakhir adalah membuat instance di setiap kelas dalam hierarki.

Adapun struktur ontologinya adalah sebagai berikut:

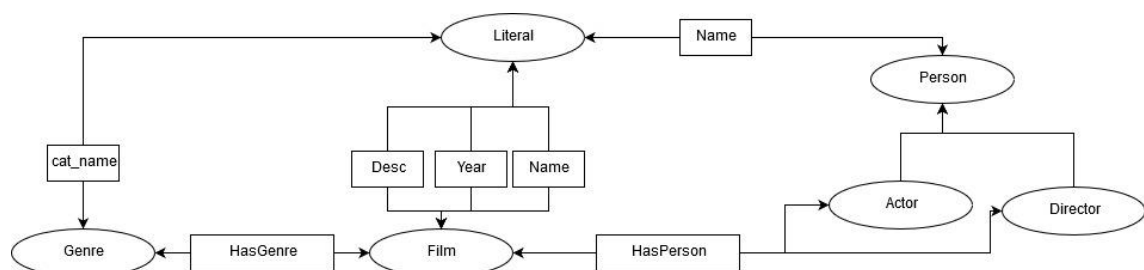
- a. Class Person terdiri dari:
 - Aktor, berisi datatype nama.
 - Direktur, berisi datatype nama.
- b. Class Film, berisi datatype judul, desc, dan tahun.
- c. Class Genre berisi datatype nama genre

Dari kelas dan datatype yang sudah didefinisikan, kemudian diperlukan suatu relasi yang menghubungkannya. Class bertindak sebagai subjek ontologi dan datatype merupakan objek dalam ontologi. Predikat dalam ontologi didefinisikan sebagai penghubung antara subjek dan objek agar memiliki arti kesatuan. Adapun predikat dari ontologi merupakan objek ontologi, dimana bentuknya seperti tabel berikut ini.

Domain	Object Ontology	Range
Film	SutradaraOleh	Director
Film	AktorOleh	Actor
Film	HasGenre	Genre

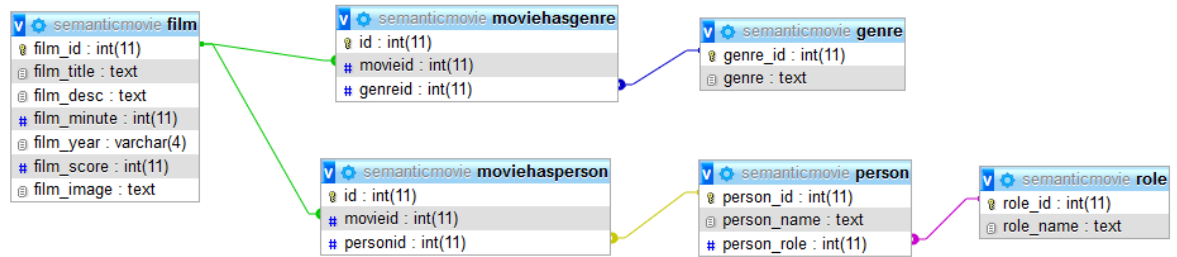
Tabel 4.1 Contoh sintaks dari RDF

Hasil ontologi yang telah dibuat ditunjukkan dalam Grafik RDF di bawah ini. Gambar tersebut menunjukkan kelas ontologi yang telah dibuat bersama dengan properties yang dimiliki oleh masing-masing kelas.



Gambar 4.4 Ontologi

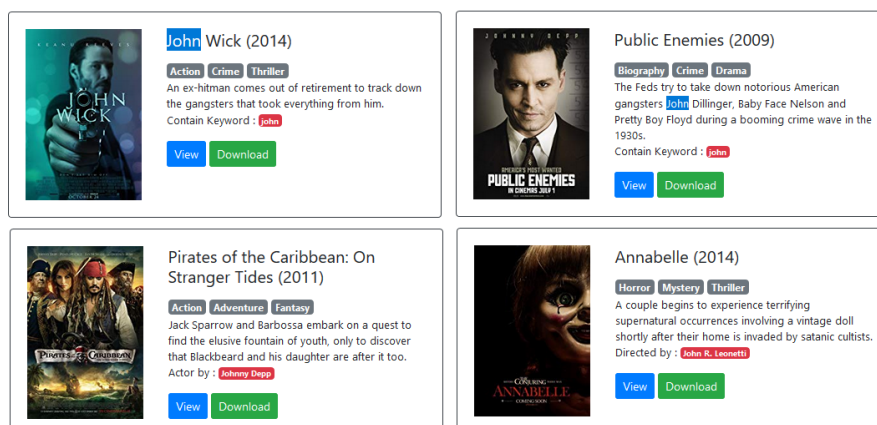
Dari hasil ontologi diatas maka dijadikan dasar untuk pembentukan tabel pada database sebagai berikut.



Gambar 4.5 Tabel Database

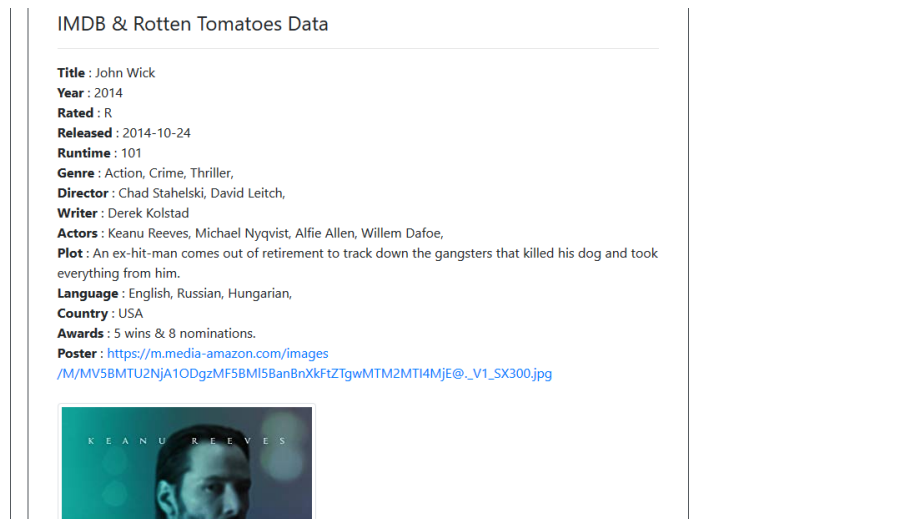
Setelah membuat tabel pada database kemudian selanjutnya adalah memasukkan data ke dalam data store, yaitu MySQL, dimana akan dilakukan simulasi query User memasukan query pencarian ke dalam sistem, kemudian sistem akan menampilkan hasil dengan urutan

1. Judul/Deskripsi mengandung kata kunci semisal user menginputkan kata “john” maka akan keluar film yang berjudul John Wick(2014), John Carter (2012) atau Live Free or Die Hard (2007) yang mengandung deskripsi “John McClane and a young hacker join forces to take down master cyber-terrorist Thomas Gabriel in Washington D.C.”
2. Dimainkan oleh actor yang sesuai kata kunci semisal user menginputkan kata “john” maka akan keluar film yang berjudul Public Enemies (2009) atau Pirates of the Caribbean: At World's End (2007) yang dibintangi oleh Johnny Depp
3. Disutradarai oleh director yang sesuai kata kunci semisal user menginputkan kata “john” maka akan keluar film yang berjudul Annabelle (2014) yang disutradarai oleh “John R. Leonetti”

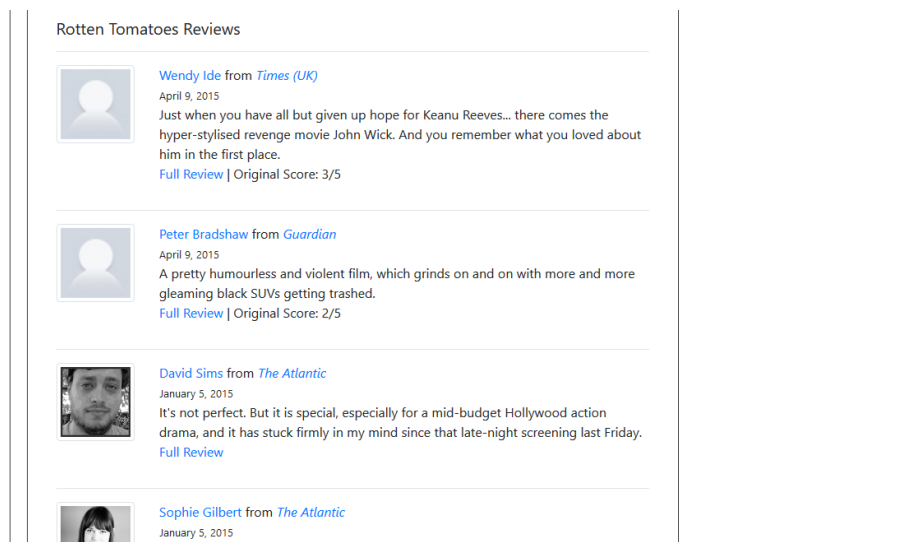


Gambar 4.6 Urutan Query

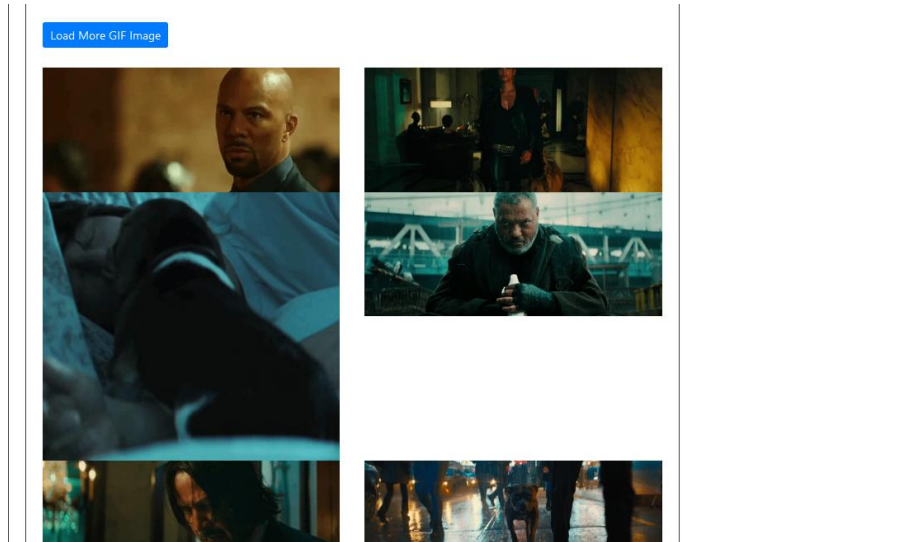
Ketika user memilih tombol download maka sistem mengarahkan user ke situs download film [Pahe.in](#). Ketika user memilih tombol view maka akan menampilkan detail dari film tersebut. Detail dari film didapat dari database lokal dan API request menggunakan [OMDb API](#). Dari OMBd API didapat alamat [Rotten Tomatoes](#) yang digunakan untuk menampilkan review film menggunakan [PHP SIMPLE HTML DOM](#). Selain menampilkan detail film sistem juga menampilkan gambar gif yang terkait dari situs [Giphy](#)



Gambar 4.7 Hasil API OMDB



Gambar 4.8 Hasil Review Rotten Tomatoes



Gambar 4.8 Hasil API Giphy

BAB V

PENUTUP

5.1.Kesimpulan

Semantic web atau web semantik merupakan salah satu perkembangan pada aplikasi web. Menurut bahasa, web semantik mempunyai arti web yang memiliki makna. Standar mempromosikan format data umum dan protokol pertukaran di Web, yang paling fundamental adalah Resource Description Framework (RDF). Dengan kata lain, web semantik merupakan suatu aplikasi web yang mempunyai knowledge base tertentu sehingga bisa dikatakan web semantik mempunyai sifat lebih pintar dari web sebelumnya.

Website MovieDatabase merupakan suatu website yang menerapkan konsep web semantik. MovieDatabase menggunakan bahasa ontologi atau OWL yang terdiri dari beberapa RDF untuk merepresentasikan informasi – informasi tentang film bioskop. Langkah – langkah yang dikerjakan dalam membuat ontologi dalam MovieDatabase meliputi menentukan domain, mempertimbangkan penggunaan ontologi yang sudah ada, menentukan istilah penting dalam ontologi, mendefinisikan dan menyusun class ontologi, mendefinisikan slot atau property, serta membuat instances. MovieDatabase juga dilengkapi fitur detail informasi dari IMDB, Review dari Rotten Tomatoes dan gambar terkait di situs Giphy

5.2.Saran

- Perlu ditambahi fitur untuk film yang berkaitan dengan film yang sedang dibuka oleh pengguna dengan menggunakan teknik ontologi.
- Fitur filtering pada website MovieDatabase perlu dikembangkan agar mampu melakukan filtering berdasarkan rating, kategori dan sebagainya.

LAMPIRAN

Index.php

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1, shrink-to-fit=no">
    <meta name="description" content="Tugas Semantik Web">
    <meta name="keywords" content="Movie Database IMDB" />
    <meta name="author" content="Sabri Sangjaya">
    <title>Movie Database</title>
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <script src="js/jquery.min.js"
type="text/javascript"></script>
    <script src="js/popper.min.js"
type="text/javascript"></script>
    <script src="js/bootstrap.min.js"
type="text/javascript"></script>
    <link rel="stylesheet" href="css/font-awesome.css">
  </head>
  <body>
    <br/><br/>
    <div class="container">
      <div class="row">
        <div class="col-md-12">
          <div class="card">
            <div class="card-body">
              <?php include 'header.php'; ?>
              <br/>
              <?php include 'searchbrowse.php'; ?>
            </div>
          </div>
        <br/>
        <br/>
      </div>
    </div>
  </body>
</html>
```

search.php

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1, shrink-to-fit=no">
    <meta name="description" content="Tugas Semantik Web">
    <meta name="keywords" content="Movie Database IMDB" />
    <meta name="author" content="Sabri Sangjaya">
    <title>Movie Database</title>
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <script src="js/jquery.min.js"
type="text/javascript"></script>
    <script src="js/popper.min.js"
type="text/javascript"></script>
    <script src="js/bootstrap.min.js"
type="text/javascript"></script>
```

```

<link rel="stylesheet" href="css/font-awesome.css">
</head>
<body>
<br/><br/>
<div class="container">
<div class="row">
<div class="col-md-12">
<div class="card border-dark">
<div class="card-body">
<?php include 'header.php'; ?>
<br/>
<?php include 'searchbrowse.php'; ?>
<br/><br/>
<?php
if(isset($_GET['q'])) {
$q=$_GET['q'];
echo "<h3>Search Result for : ";
echo $q;
echo "</h3>";
}?>
<hr/>
<div class="row">
<?php
include("proses/connect.php");
if(isset($_GET['q'])) {
$q=$_GET['q'];
$sql="SELECT 1 as resultquery,film.*,
group_concat(DISTINCT genre.genre) as genre,
group_concat(DISTINCT person_name) as actor FROM `film` JOIN
moviehasgenre on film.film_id=moviehasgenre.movieid JOIN
moviehasperson on film.film_id=moviehasperson.movieid JOIN
person on person.person_id=moviehasperson.personid join genre on
genre.genre_id=moviehasgenre.genreid WHERE film_title LIKE
'%" . $q . "%' OR film_desc LIKE '%" . $q . "%' AND person.person_role=2
group by film_id UNION SELECT 2 as resultquery,film.*,
group_concat(DISTINCT genre.genre) as genre,
group_concat(DISTINCT person_name) as actor FROM `film` JOIN
moviehasgenre on film.film_id=moviehasgenre.movieid JOIN
moviehasperson on film.film_id=moviehasperson.movieid JOIN
person on person.person_id=moviehasperson.personid join genre on
genre.genre_id=moviehasgenre.genreid WHERE person_name LIKE
'%" . $q . "%' and person.person_role=2 group by film_id UNION
SELECT 3 as resultquery,film.*, group_concat(DISTINCT
genre.genre) as genre, group_concat(DISTINCT person_name) as
actor FROM `film` JOIN moviehasgenre on
film.film_id=moviehasgenre.movieid JOIN moviehasperson on
film.film_id=moviehasperson.movieid JOIN person on
person.person_id=moviehasperson.personid join genre on
genre.genre_id=moviehasgenre.genreid WHERE person_name LIKE
'%" . $q . "%' and person.person_role=1 group by film_id order by
(case when film_title LIKE '%" . $q . "%' then 1 when film_desc LIKE
'%" . $q . "%' then 2 else 3 end)";
$query = mysqli_query($conn, $sql);
if($query!=null&&mysqli_num_rows($query)>0) {
while($data = mysqli_fetch_array($query)) {
?>
<div class="col-md-6" style="margin-bottom:
10px;">
<div class="card border-dark">
<div class="card-body">
<div class="row">

```

```

" alt="poster">

<div class="col-sm-8">
<h5 class="card-title"><?php echo
$data['film_title']. " (". $data['film_year']. ")";?></h5>
<?php
$array=explode(",",$data['genre']);
foreach ($array as $item) {
echo "<a href='#' class='badge badge-
secondary'>$item</a> ";
}
?>
<p class="card-text" style="font-size:
0.8em"><?php echo $data['film_desc'];?>

<?php
if($data['resultquery']==2){
echo "<br/>Actor by : ";
$array=explode(",",$data['actor']);
foreach ($array as $item) {
echo "<a href='#' class='badge badge-
danger'>$item</a> ";
}
}
else if($data['resultquery']==3){
echo "<br/>Directed by : ";
$array=explode(",",$data['actor']);
foreach ($array as $item) {
echo "<a href='#' class='badge badge-
danger'>$item</a> ";
}
}
else if($data['resultquery']==1){
echo "<br/>Contain Keyword : ";
echo "<a href='#' class='badge badge-
danger'>$q</a> ";
}
?>
</p>
<a href="detail.php?id=<?php echo
$data['film_id']?>" class="btn btn-primary btn-sm">View</a>
<a href="https://pahe.in/?s=<?php echo
$data['film_title']. ' '. $data['film_year']. ''?>" target="_blank"
class="btn btn-success btn-sm">Download</a>
</div>
</div>
</div>
</div>
<?php
}
}else{
echo"Not found";
}
}
?>

```

```

<?php
include("proses/connect.php");
if(isset($_GET['cat'])){
    $q=$_GET['cat'];
    $sql="SELECT film.*, group_concat(DISTINCT
genre.genre) as genre, group_concat(DISTINCT
person_name,person.person_role) as actor FROM `film` JOIN
moviehasgenre on film.film_id=moviehasgenre.movieid JOIN
moviehasperson on film.film_id=moviehasperson.movieid JOIN
person on person.person_id=moviehasperson.personid join genre on
genre.genre_id=moviehasgenre.genreid WHERE genre.genre='".$q."'
group by film_id order by film.film_score DESC";
    //echo $sql;
    $query = mysqli_query($conn, $sql);
    if($query!=null&&mysqli_num_rows($query)>0){
        while($data = mysqli_fetch_array($query)) {
            ?>
            <div class="col-md-6" style="margin-bottom:
10px;">
                <div class="card border-dark">
                    <div class="card-body">
                        <div class="row">
                            " alt="poster">
                            <div class="col-sm-8">
                                <h5 class="card-title"><?php echo
$data['film_title']." (".$data['film_year'].")";?></h5>
                                <?php
                                $array=explode(",",$data['genre']);
                                foreach ($array as $item) {
                                    echo "<a href='#' class='badge badge-
secondary'>$item</a> ";
                                }
                                ?>
                                <p class="card-text" style="font-size:
0.8em"><?php echo $data['film_desc'];?>
                                <?php
                                $arrayactor=[];
                                $arraydirector=[];
                                $array=explode(",",$data['actor']);
                                foreach ($array as $key => $item) {
                                    if (substr($item, -1)==2){

                                        array_push($arrayactor,substr($item,0,-1));
                                    }
                                    if (substr($item, -1)==1){

                                        array_push($arraydirector,substr($item,0,-1));
                                    }
                                }
                                echo "<br/><b>Actor : </b>";
                                foreach ($arrayactor as $key =>
$item) {
                                    echo "<a href='#' class='badge
badge-success'>$item</a> ";
                                }
                                echo "<br/><b>Director : </b>";

```



```
<meta name="viewport" content="width=device-width, initial-
scale=1, shrink-to-fit=no">
<meta name="description" content="Tugas Semantik Web">
<meta name="keywords" content="Movie Database IMDB" />
<meta name="author" content="Sabri Sangjaya">
<title><?php echo $judul." - Movie Database"?></title>
<link rel="stylesheet" href="css/bootstrap.min.css">
<script src="js/jquery.min.js"
type="text/javascript"></script>
<script src="js/index.js" type="text/javascript"></script>
<script src="js/popper.min.js"
type="text/javascript"></script>
<script src="js/bootstrap.min.js"
type="text/javascript"></script>
<link rel="stylesheet" href="css/font-awesome.css">
</head>
<body>
<br/><br/>
<div class="container">
<div class="row">
<div class="col-md-12">
<div class="card border-dark">
<div class="card-body">
<?php include 'header.php'; ?>

<hr/>
<div class="row">
<div class="col-md-9" style="margin-bottom:
10px;">

<div class="card border-dark">
<div class="card-body">
<div class="row">

" alt="poster">

<div class="col-sm-8">
<h3 class="card-title"
id="judul"><?php echo $data['film_title']?>
</h3><h3><?php echo
"(".$data['film_year'].")";?></h3>
<?php
$array=explode(",",$data['genre']);
foreach ($array as $item) {
echo "<a href='#' class='badge badge-
secondary'>$item</a> ";
}
?>
<p class="card-text" style="font-size:
0.8em"><?php echo $data['film_desc'];?>
<?php
$arrayactor=[];
$arraydirector=[];
$array=explode(",",$data['actor']);
foreach ($array as $key => $item) {
if (substr($item,-1)==2){
```

```

array_push($arrayactor, substr($item, 0, -1));
    }
    if (substr($item, -1) == 1) {

array_push($arraydirector, substr($item, 0, -1));
    }
    }
    echo "<br/><b>Actor : </b>";
    foreach ($arrayactor as $key =>

$item) {
        echo "<a href='#' class='badge
badge-success'>$item</a> ";
    }
    echo "<br/><b>Director : </b>";
    foreach ($arraydirector as $key =>

$item) {
        echo "<a href='#' class='badge
badge-warning'>$item</a>";
    }
    ?>
</p>
    <a href="https://pahe.in/?s=<?php echo
$data['film_title'].' '.$data['film_year'].'?>" target="_blank"
class="btn btn-success btn-sm">Download</a>
</div>
</div>
<br/>
<div class="rows">
    <div class="col-md-12">
        <h4>IMDB & Rotten Tomatoes Data</h4>
        <hr/>
        <?php
        try {
            $omdb = new OMDb();
            $omdb->setParams( ['tomatoes' => TRUE,
'plot' => 'full', 'apikey' => 'trilogy'] );
            $omdb->setParam( 'y',
$data['film_year'] );
            $movie = $omdb->get_by_title(
$data['film_title'] );
            foreach ($movie as $key => $value) {
                if(is_array($value)){
                    echo "<b>".$key."</b> : ";
                    foreach ($value as $key2 => $value2) {
                        if(is_array($value2)){
                            echo "<br/>";
                            foreach ($value2 as $key3 => $value3)
                            {
                                echo "<b>".$key3."</b> :
                                ".$value3."<br/>";
                            }
                        }else{
                            echo $value2.", ";
                        }
                    }
                    echo "<br/>";
                }else{
                    if(is_null($value)){
                    }
                    else if($key=="Response"){

```

```

    }
    else
    if ($key=="Poster" || $key=="tomatoURL") {
        $salamattomato=$movie["tomatoURL"];
        echo "<b>".$key."</b> : <a
target='_blank' href='\".$value.\"'>\".$value.\"</a><br/>";
        if ($key=="Poster") {
            echo "<br/><img src='\".$value.\"'
alt='poster' class='img-fluid img-thumbnail'><br/>";
        }
    }
    else{
        echo "<b>".$key."</b> :
\".$value.\"<br/>";
    }
}
}
}
} catch (Exception $e) {
    echo $e->getMessage();
}
?>
<br/>
<h5>Rotten Tomatoes Reviews</h5>
<hr/>
<div class="review_table">
<?php
// Create DOM from URL or file

$salamattomato.="reviews?type=top_critics";
$html = file_get_html($salamattomato);
try {
    foreach ($html->find('div[class="row
review_table_row"]') as $element){
        $tmt_gambar=$element-
>find('img.critic_thumb', 0)->src;
        $tmt_nama= $element-
>find('div.critic_name', 0)->find('a', 0)->innertext;
        $tmt_namaurl= $element-
>find('div.critic_name', 0)->find('a', 0)->href;
        $tmt_namal= $element-
>find('div.critic_name', 0)->find('a', 1)->innertext;
        $tmt_namalurl= $element-
>find('div.critic_name', 0)->find('a', 1)->href;
        $tmt_review= $element-
>find('div.the_review', 0)->innertext;
        $tmt_reviewlink=$element-
>find('div.review-link', 0)->innertext;
        $tmt_reviewtgl= $element-
>find('div.review-date', 0)->innertext;
        ?>
        <div class="row">
            <div class="col-md-2"></div>
            <div class="col-md-10">
                <a href="<?php echo
"https://www.rottentomatoes.com\".$tmt_namaurl;?>"
target="_blank"><?php echo $tmt_nama;?></a> from
                <a href="<?php echo
"https://www.rottentomatoes.com\".$tmt_namalurl;?>"
target=" blank"><?php echo $tmt_namal;?></a><br/>

```



```
<div class="input-group mb-3">
  <input type="text" name="q" class="form-control"
placeholder="Type a title or actor or director of movie">
  <div class="input-group-append">
    <button class="btn btn-success" type="submit">Go</button>
  </div>
</div>
</form>
<h4>Browse by Category</h4>
<?php
include("proses/connect.php");
$sql="SELECT * FROM `genre` WHERE 1";
//echo $sql;
$badge=["badge-primary","badge-secondary","badge-
success","badge-danger","badge-warning","badge-info","badge-
dark"];
$it = 0;
$query = mysqli_query($conn, $sql);
if($query!=null&&mysqli_num_rows($query)>0){
while($data = mysqli_fetch_array($query)) {
echo "<a href='search.php?cat=". $data["genre"]."' class='badge
". $badge[$it]."'>". $data["genre"]."</a> ";
$it++;
if($it==count($badge)){
$it=0;
}
?>
<?php
}
}
?>
```