PENERAPAN TEKNIK WEB SCRAPING PADA SITUS IMDb DENGAN NODE JS

Arif Maulana Komarudin[1)], Asep Nurul Huda[2)], Doni Agistira[3)]

[1,2,3]Program Studi Informatika Fakultas Teknik Universitas Siliwangi

e-mail: 177006085@student.unsil.ac.id, 177006063@student.unsil.ac.id, 177006075@student.unsil.ac.id

Abstrak

Web Scraping merupakan salah satu cara untuk ekstraksi konten halaman situs. Melalui teknik ini data dapat diambil dari suatu situs tanpa harus membuka situs tersebut menggunakan browser. Internet Movie Database (https://www.imdb.com/) merupakan suatu situs yang menyediakan informasi terkait: film, acara TV, dan lainnya. Produk dan layanan IMDb dirancang untuk dapat diakses melalui situs web, perangkat seluler dan IMDb X-Ray pada perangkat Fire TV serta menawarkan saluran streaming gratis.Tujuan dari penelitian ini melakukan pengambilan data dari web target dengan menerapkan scraping. Html parsing dipilih sebagai metode scraping yang akan digunakan dalam percobaan pada penelitian ini, serta, penggunaan node.js dengan tambahan modul Cheerio. Percobaan pada penelitian ini telah berhasil mengimplementasikan teknik scraping dan mengambil data dari web target.

Kata Kunci : html parser, node.js, web scraping

Abstract

Web Scraping is one way to extract site page content. Through this technique data can be retrieved from a site without having to open the site using a browser. Internet Movie Database (https://www.imdb.com/) is a site that provides related information: movies, TV shows, and more. IMDb products and services are designed to be accessible via websites, mobile devices and IMDb X-Ray on Fire TV devices and offer free streaming channels. The aim of this research is to retrieve data from the target web by applying scraping. Html parsing was chosen as the scraping method to be used in the experiments in this study, as well as the use of node.js with the addition of the Cheerio module. Experiments in this study have succeeded in implementing scraping techniques and retrieving data from the target web.

***Keywords** : html parser, node.js, web scraping*

I. PENDAHULUAN

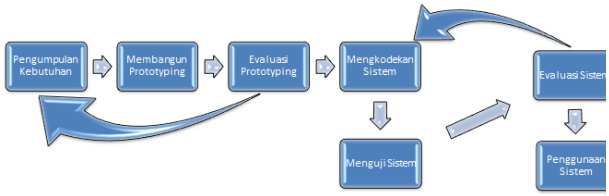
Salah satu cara untuk memisahkan konten utama halaman situs dengan bagian-bagian yang tidak berhubungan dengan isi adalah dengan menggunakan teknik scraping[1]. Melalui teknik ini, informasi atau data dapat diambil dari suatu situs tanpa harus masuk ke dalam situs tersebut serta data yang diambil dapat terjamin kesamaanya.

Salah satu situs yang menyediakan beragam informasi serta dapat diekstraksi informasinya adalah situs Internet Movie Database (IMDb). IMDb merupakan salah satu sumber informasi terkait film, acara TV, dan selebriti. Produk dan layanan untuk membantu penggemar memutuskan apa yang harus ditonton dan di mana menontonnya termasuk situs web dan perangkat seluler dan IMDb X-Ray pada perangkat Fire TV. IMDb juga menawarkan saluran streaming gratis [2].

Kelebihan dari IMDb adalah informasi yang lengkap serta dukungan komunitas yang besar sehingga informasi yang dihasilkan menjadi lebih dipercaya. Teknik Scraping dapat dilakukan dengan berbagai cara, salah satunya adalah dengan html parsing. Ini merupakan cara yang umum digunakan, serta banyak juga alat bantu yang dapat digunakan seperti Node Js dengan tambahan modul Cheerio yang *open source*. Sehingga dapat dipergunakan dengan seluas luasnya.

II. METODOLOGI

Penelitian ini menggunakan model *protyping*, sistem dikembangan secara bertahap. Setiap tahap pengembangan dilakukan percobaan-percobaan untuk melihat apakah sistem sudah bekerja sesuai dengan yang diinginkan. Terdapat 6 fase yang dilakukan: analisa kebutuhan, membangun *prototyping*, evaluasi *prototyping*, *m*engkodekan sistem, menguji sistem, evaluasi sistem, seperti ditampilkan pada gambar 1[3].



Gambar 1. Skema Model *prototyping*

III. TINJAUAN PUSTAKA

1. Web Scraping

Web Scraping adalah proses yang melibatkan pengambilan dokumen semi-terstruktur dari internet, umumnya laman web dalam bahasa markup seperti HTML atau XHTML. Analisis dokumen untuk mengekstraksi data tertentu dari suatu halaman web dikenal sebagai *screen scraping[4]*. *Web Scraping* berbeda dengan *data mining* karena dalam data mining mengimpilkasikan cara untuk melihat pola semantik dalam jumlah data berukuran yang sudah ada.

Gambar 2. Ilustrasi Cara Kerja Web Scrapper

2. HTML Parsing

HTML parsing adalah proses menganalisis serangkaian simbol, dalam bahasa alami atau dalam bahasa komputer, sesuai dengan aturan tata bahasa formal. Parsing HTML mencakup proses ekstraksi dan memproses informasi yang relevan seperti judul kepala, aset halaman, bagian utama dan kemudian, menyimpan file yang diproses[5].

3. Node JS

Node JS merupakan platform perangkat lunak pada sisi server dan aplikasi jaringan. Ditulis dengan bahasa JavaScript dan dijalankan pada Windows, Mac OS X, dan Linux tanpa perubahan kode program. Node.js memiliki pustaka peladen HTTP sendiri sehingga memungkinkan untuk menjalankan peladen web tanpa menggunakan program peladen web seperti Apache atau Lighttpd.[6]

4. Cheerio JS

Cheerio *js* merupakan sebuah modul dari *Node JS* untukImplementasi inti jQuery cepat, fleksibel, dan ramping yang dirancang khusus untuk server.[7]

IV. HASIL DAN PEMBAHASAN

1. Analisis Kebutuhan

Sistem yang dibuat bertujuan untuk mendapatkan informasi film tanpa harus masuk kedalam situs IMDb tersebut. Hal-hal yang diharapkan dari pengguna agar dapat diwujudkan dalam sistem ini diantaranya adalah sebagai berikut :

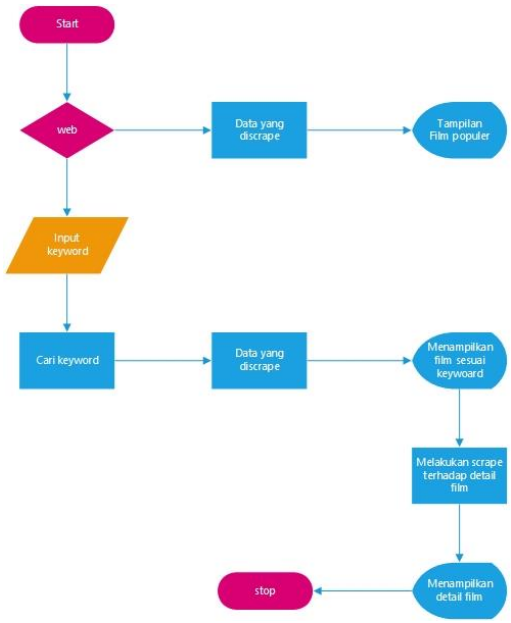
a) Sistem dapat secara otomatis mengekstrak informasi film dari situs IMDb

b) Sistem dapat secara otomatis dapat menampilkan hasil scraping

c) Membuat portal website yang dapat menampilkan data hasil scraping dari situs utama.

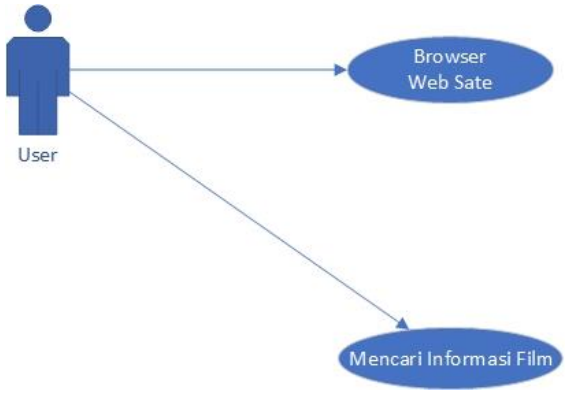
2. Membangun Prototype

Sebelum membangun prototype maka dibuatkan flowchart sistem serta use case diagram. Flowchart sistem yang dibuat ditampilkan pada gambar 3.



Gambar 3. Flowchart Sistem

Pada gambar 3 ditampilkan tahapan user mencari informasi sebelum melakukn scraping data. Pertama user membuka aplikasi web kemudian sistem akan melakukan scraping dari situs IMDb, data kemudian akan ditampilkan, ketika user ingin mencari film dengan keywoard tertentu maka sistem akan mencari data berdasarkan keywoard yang dimasukan, lalu jika user mengklik detail film maka sistem akan melakukan scraping terhadap film tersebut kemudian menampilkanya.



Gambar 3. Use Case Diagram

Pada gambar 3 terlihat uraian kegiatan user dalam melakukan pencarian data film, tidak terdapat admin karena tidak menggunakan database dan hanya menampilkan data yang diambil.

3.3 Evaluasi *Prototyping*

Desain protetype tidak mengalami perubahan.

3.4 Mengkodekan Sistem

Pada tahap ini rancangan diimplementasikan ke dalam suatu program komputer menggunakan bahasa pemrograman. Potongan kode program untuk proses scraping ditampikan pada gambar 4.

```
const fetch = require("node-fetch");
const cheerio = require("cheerio");

const searchUrl =
"https://www.imdb.com/find?s=tt&ttype=ft&ref_=fn
_ft&q=";
const movieUrl = "https://www.imdb.com/title/";
const indexUrl =
"https://www.imdb.com/chart/moviemeter/?ref_=nv_
mv_mpm";

const searchCache = {};
const movieCache = {};

function popularMovie() {
  return fetch(`${indexUrl}`)
    .then(response => response.text())
    .then(body => {
       const indexs = [];
       const $ = cheerio.load(body);
            $('.lister-list tr').each(function
(i, element) {
    const $element = $(element);
                const $image =
$element.find('.posterColumn a img');
                const $title =
$element.find('.titleColumn a');
                const $rating =
$element.find('.ratingColumn strong');
                const imdbID =
$title.attr('href').match(/title\/(.*)\//)[1];
     const index = {imdbID,image:
$image.attr("src").replace("._V1_UY67_CR0,0,45,6
7_AL_.jpg",
"._V1_SY1000_CR0,0,670,1000_AL_.jpg"),
                    title: $title.text(),
                    rating: $rating.text()
                };
                indexs.push(index);
     });
            return indexs;
  })
}

function searchMovies(searchTerm) {
    if (searchCache[searchTerm]) {
        console.log('Serving from cache:',
searchTerm);
        return
Promise.resolve(searchCache[searchTerm]);
    }
    return fetch(`${searchUrl}${searchTerm}`)
        .then(response => response.text())
        .then(body => {
            const movies = [];
            const $ = cheerio.load(body);
            $('.findResult').each(function (i,
element) {
                const $element = $(element);
                const $image = $element.find('td
a img');
                const $title =
$element.find('td.result_text a');

                const imdbID =
$title.attr('href').match(/title\/(.*)\//)[1];

                const movie = {
                    image:
$image.attr("src").replace("._V1_UX32_CR0,0,32,4
4_AL_.jpg",
"._V1_SY1000_CR0,0,670,1000_AL_.jpg"),
                    title: $title.text(),
                    imdbID
                };
                movies.push(movie);
            });
            searchCache[searchTerm] = movies;

            return movies;
        });
}

function getMovie(imdbID) {
    if (movieCache[imdbID]) {
        console.log('Serving from cache:',
imdbID);
        return
Promise.resolve(movieCache[imdbID]);
    }
    return fetch(`${movieUrl}${imdbID}`)
        .then(response => response.text())
        .then(body => {
            const $ = cheerio.load(body);
            const $title = $('.title_wrapper
h1');

            const title =
$title.first().contents().filter(function () {
                return this.type === 'text';
            }).text().trim();
            const imdbRating =
$('span[itemProp="ratingValue"]').text();
            const poster = $('div.poster a
img').attr('src').replace("._V1_UY268_CR0,0,182,
268_AL_.jpg",
```

P-ISSN 2477-3891
E-ISSN 2615-4765

```
"._V1_SY1000_CR0,0,670,1000_AL_.jpg");
            const summary =
$('div.summary_text').text().trim();

            const movie = {
                imdbID,
                title,
                imdbRating,
                poster,
                summary
            }

            movieCache[imdbID] = movie;
            return movie;
        });
}

module.exports = {
    searchMovies,
    getMovie,
    popularMovie
}
```

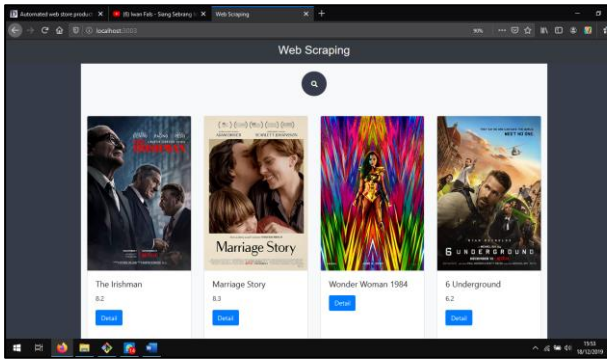
Gambar 4. Potongan Source Code Scraping Web

3.5 Menguji Sistem

Setelah sistem dibuat maka langkah selanjutnya adalah menguji sistem apakah berjalan sesuai dengan rencana atau tidak.

3.5.1 Halaman Awal

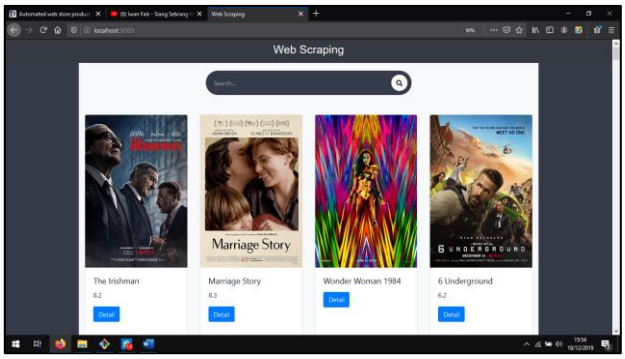
Pada saat aplikasi *web scraping* dijalankan, maka akan menampilkan halaman depan yang berisi hasil scraping dari situs IMDb seperti ditampilkan pada gambar 5.



Gambar 5. Tampilan Aplikasi

3.5.2 Menu Pencarian

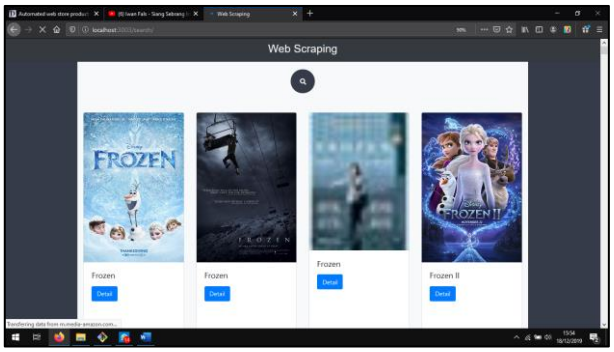
Pada halaman utama, di bagian atas terdapat menu untuk mencari film. Jika data yang dicari, terdapat pada data hasil scraping maka aplikasi akan menampilkanya seperti pada gambar 6.



Gambar 6. Tampilan Menu cari

3.5.3. Halaman Hasil Pencarian

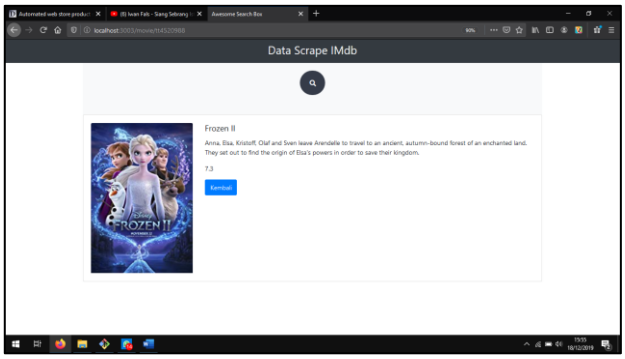
Data hasil pencarian yang telah ditemukan akan ditampilkan dalam aplikasi, seperti ditampilkan pada gambar 7.



Gambar 7. Hasil Pencarian

3.5.4 Halaman Detail

Fitur ini dirancang untuk menampilkan informasi detail terkiat film yang dipilih oleh user, seperti ditampilkan pada gambar 8.



Gambar 8. Halaman Detail Film

3.6 Evaluasi Sistem

Percobaan scraping data pada web target berhasil dilakukan. Data hasil scraping dapat digunakan atau ditampilkan melalui web, sehingga mudah untuk diakses.

V. KESIMPULAN

Web scraping dengan teknik html parsing berhasil digunakan dalam percobaan pada penelitian ini, serta penggunaan node.js dengan tambahan modul Cheerio. Data hasil scraping dapat digunakan lagi untuk tujuan lain, atau ditampilkan melalui web browser agar lebih mudah diakses.

DAFTAR PUSTAKA

[1] M. S. Utomo, "No Title," *J. Teknol. Inf. Din. Vol.*, vol. 17, no. 2, pp. 147–153, 2012.
[2] "Ruang Pers - IMDb." [Online]. Available: https://www.imdb.com/pressroom/?ref_=helpms_ih_gi_whatsimdb. [Accessed: 18-Dec-2019].
[3] "Mengenal Prototyping - DOT Intern - Medium." [Online]. Available: https://medium.com/dot-intern/sdlc-metode-prototype-8f50322b14bf. [Accessed: 18-Dec-2019].
[4] M. Turland, "No Title," in *php|architect's Guide to Web Scraping with PHP*, 2010, p. 2.
[5] "Developer Tools - Open-Source HTML Parser." [Online]. Available: https://blog.appseed.us/developer-tools-html-parser/. [Accessed: 18-Dec-2019].
[6] "Node.js - Wikipedia bahasa Indonesia, ensiklopedia bebas." [Online]. Available: https://id.wikipedia.org/wiki/Node.js. [Accessed: 18-Dec-2019].
[7] "cheerio | Implementasi inti jQuery cepat, fleksibel, dan ramping yang dirancang khusus untuk server." [Online]. Available: https://cheerio.js.org/. [Accessed: 18-Dec-2019].