

# Implementasi *Scraping Google Scholar* Menggunakan HTML DOM Untuk Pengumpulan Data Artikel Dosen UPN Veteran Jakarta Berbasis Web

Azini Fauzia Putri<sup>1</sup>, Geyessella Manik<sup>2</sup>, Farhana Nabila<sup>3</sup>, Nurul Chamidah<sup>4</sup>  
Informatika / Fakultas Ilmu Komputer  
Universitas Pembangunan Nasional Veteran Jakarta  
Jl. Rs. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12450, Indonesia  
azinifp@upnvj.ac.id<sup>1</sup>, geyessella.manik@gmail.com<sup>2</sup>, farhananabila@upnvj.ac.id<sup>3</sup>,  
nurul.chamidah@upnvj.ac.id<sup>4</sup>

**Abstrak.** Google Cendekia atau biasa disebut *Google Scholar* adalah situs pencarian literatur yang berdiri pada tahun 2004. Bagi civitas academica terutama bagi dosen UPN Veteran Jakarta, melakukan publikasi penelitian merupakan hal yang penting. Untuk menilai kinerja penelitian diperlukan suatu *website* yang berisi khusus publikasi artikel ilmiah para dosen yang dapat melakukan rekapitulasi publikasi artikel yang telah disitasi dari *Google Scholar*. Pemilihan situs *Google Scholar* karena data karya ilmiah telah terdokumentasi secara lengkap. Adapun proses pengembangan web dilakukan dengan menggunakan model *classic life cycle* atau biasa disebut dengan *waterfall* yang terdiri atas empat langkah yaitu analisa keperluan sistem, desain, penyusunan kode, dan terakhir adalah pengujian. Penarikan data publikasi sendiri menggunakan teknik *web scraping*, teknik ini mempermudah dalam penarikan data-data publikasi dalam jumlah yang besar secara cepat. *Scraping* dilakukan dengan menggunakan bahasa pemrograman PHP dengan bantuan *library* Simple HTML DOM Parser yang berhasil mengambil 352 data dosen dari 7 fakultas UPN Veteran Jakarta dengan banyaknya data artikel sebesar 4.977 data artikel dosen. Data tersebut akan disimpan ke dalam *database* dan ditampilkan pada *website* yang berfungsi untuk menampilkan daftar profil dosen, judul karya ilmiah, tahun dan sitasi yang dilengkapi fitur pencarian dan *sorting* data sehingga memudahkan pengguna untuk melihat jumlah sitasi jurnal dan informasi lainnya.

**Kata Kunci:** *Google Scholar*, *Scraping*, HTML DOM.

## 1 Pendahuluan

Perkembangan data semakin pesat seiring berjalannya waktu sehingga membuat orang semakin cepat dalam mencari dan mengakses informasi. Salah satu data yang ada di internet yaitu data publikasi hasil penelitian. Bagi peneliti civitas academica pada perguruan tinggi, melakukan publikasi penelitian merupakan hal yang penting. Bentuk publikasi dapat meliputi buku, jurnal, dan artikel ilmiah. Karya-karya ilmiah tersebut bisa didapatkan dengan mudah menggunakan akses mesin pencari khusus bernama *Google Scholar*. *Google Cendekia* adalah salah satu situs pencarian jurnal yang diluncurkan pada tahun 2004 [1].

Dengan adanya *Google Cendekia* atau sering dikenal dengan sebutan *Google Scholar*, pengguna dapat melakukan eksplorasi materi pelajaran dalam bentuk teks yang bisa dipakai menjadi tempat dalam mencari literatur akademis berupa tesis, buku, abstrak, dan artikel. *Google Scholar* dapat menampilkan semua publikasi hasil penelitian dari berbagai *website*, data karya ilmiah pada scholar juga terdokumentasikan secara lengkap.

Melakukan publikasi jurnal ilmiah adalah hal yang penting bagi semua akademisi terutama bagi dosen dan perguruan tinggi UPN Veteran Jakarta. Data publikasi dengan jumlah yang besar tentunya akan mempersulit dalam melakukan rekapitulasi secara manual, karena itu maka akan dilakukan perancangan suatu *website* yang berisi data dosen UPN Veteran Jakarta dan publikasi yang membantu dalam rekapitulasi data artikel ilmiah seperti jumlah artikel, sitasi dan sebagainya dengan memanfaatkan *scraping* dalam proses penarikan informasi dari situs *Google Scholar*.

Saat ini penggunaan *data scraping* merupakan hal yang umum digunakan oleh perusahaan atau individu untuk kepentingan tertentu. *Data scraping* memiliki beberapa bentuk, salah satunya adalah *web scraping* yang berfokus pada cara dalam menarik data dari sebuah halaman web dengan ukuran data yang beragam.

*Web scraping* bekerja dengan melakukan proses ekstraksi data dengan mempelajari kode dari sebuah *website* yang hendak diambil data/informasinya, data yang ingin ditarik biasanya berbentuk teks bertipe HTML atau XHTML. Manfaat penggunaan *web scraping* yaitu dapat melakukan pengambilan data secara cepat pada jumlah data yang besar.

Bahasa pemrograman yang digunakan untuk merancang web adalah PHP dengan bantuan *library* Simple HTML DOM Parser yang membantu dalam proses penarikan data berdasarkan *id*, *class*, *tags* dan sebagainya yang ada pada kode *website* Google Scholar. HTML DOM Parser berguna untuk menemukan elemen HTML pada situs *Google Scholar* lalu menyimpan data yang berhasil ditarik pada *database* phpMyAdmin, dan akan ditampilkan pada *website* yang telah dirancang. Tujuan pembuatan *website* untuk menampilkan daftar profil dosen, judul karya ilmiah, tahun dan sitasi yang dilengkapi fitur pencarian dan *sorting* data sehingga memudahkan pengguna untuk melihat sitasi jurnal dan informasi lainnya. Judul karya ilmiah juga dapat diarahkan ke halaman karya ilmiah tersebut pada situs *Google Scholar*.

## 2 Landasan Teori

### 2.1 Google Scholar

*Google Scholar* (bahasa Indonesia: Google Cendekia) merupakan suatu fitur yang diciptakan oleh Anurag Acharya yang terikat pada tahun 2000 dengan tim *web-indexing* Google. *Google Scholar* biasanya digunakan sebagai mesin pencari bahan rujukan akademis yang membantu pengguna dalam mengidentifikasi penelitian yang paling relevan dari seluruh penelitian akademis. *Google Scholar* menyediakan layanan publikasi karya-karya penelitian seperti makalah *peer-review*, buku, thesis, abstrak, dan artikel dari penerbit akademis, komunitas, profesional, pusat informasi pracetak, universitas, dan organisasi akademis lainnya [2].

### 2.2 Web Scraping

*Web scraping* adalah suatu teknik yang mempermudah dalam hal mendapatkan informasi pada *website* secara otomatis tanpa harus mengumpulkan atau menyalin secara manual. Teknik ini dapat digunakan untuk memperoleh isi informasi sebuah halaman web secara spesifik, misalnya berupa link gambar, isi berita atau sejenisnya. Untuk mendapatkan informasi, maka memahami kode HTML dari web yang hendak diambil datanya merupakan hal penting dalam penarikan data.

Tujuan penggunaan teknik ini yaitu bagaimana cara dalam mendapatkan data/ informasi melalui penarikan dan ekstraksi data dengan *size* yang beragam yang diambil pada *website* dan digunakan bagi kepentingan lain [3]. Data yang di ambil umumnya berupa teks yang bertipe HTML atau XHTML [4]. Dalam penarikan data diperlukan file *library* `simple_html_dom.php` yang dapat diunduh dari sourceforge lalu dimasukkan ke dalam folder kode web, dan langkah selanjutnya yaitu memanggil *library* tersebut yang dimanfaatkan guna menemukan elemen HTML pada web. Kode pemanggilan *library* Simple HTML DOM dan penarikan data berdasarkan *class* dari situs *Google Scholar* dapat dilihat pada gambar 1.

```
# create and load the HTML
include('simple_html_dom.php');
$html = new simple_html_dom();
$html->load_file("http://scholar.google.se/citations?user=" . $_GET["user"]);

#find
```

```

foreach($html->find(".gsc_a_tr") as $pub) {
    $judul = trim($pub->find(".gsc_a_at", 0)->plaintext);
    $pengarang= trim($pub->find(".gs_gray", 0)->plaintext);
    $jurnal= trim($pub->find(".gs_gray", 1)->plaintext);
    if(!is_numeric($pub->find(".gsc_a_ac", 0)->plaintext))
        $citation = 0;
    else
        $citation = $pub->find(".gsc_a_ac", 0)->plaintext;
    if($pub->find(".gsc_a_h", 0)->plaintext == " ")
        $tahun_terbit =0;
    else
        $tahun_terbit = $pub->find(".gsc_a_h", 0)->plaintext;

    //echo $i.". ". $judul;

    //echo "<br/>";

    $i++;
    $scrap->create($judul,$pengarang,$tahun_terbit,$citation,$jurnal,$id_scholar);
}

```

**Gambar. 1.** Pemanfaatan Simple HTML DOM Parser.

### 2.3 phpMyAdmin

phpMyAdmin didirikan oleh Tobias Ratschiller yang merupakan sebuah aplikasi *open source* berbasis pemrograman PHP, berfungsi untuk memudahkan manajemen MySQL [4]. MySQL sendiri merupakan salah satu jenis *database* server yang menggunakan bahasa SQL dalam mengakses *database*. phpMyAdmin digunakan sebagai administrator MySQL melalui *browser* (web) untuk manajemen *database*.

Berikut beberapa fitur pada phpMyAdmin:

1. phpMyAdmin memiliki *interface* yang *user friendly*.
2. phpMyAdmin memperbolehkan *user* memanfaatkan kebanyakan fungsi MySQL, seperti membuat dan mengelola data pada *database*, menjalankan *queries* MySQL, melakukan eksekusi *statement* MySQL, mengelola *user* dan *permission*.
3. Dapat melakukan *import* dan *export* data dari dan ke berbagai format yang diinginkan.
4. Mengubah format data-data yang sudah tersimpan, dan melakukan pencarian pada *database*.

### 2.4 UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) adalah diagram terstandar industri, dapat digunakan dalam hal mendeskripsikan kebutuhan dan spesifikasi sebuah sistem. Penggunaan UML bertujuan untuk menolong tim pengembang proyek dalam berkomunikasi, mengeksplorasi dan validasi desain sistem [5].

Diagram yang terdapat di dalam UML yaitu:

1. *Use Case Diagram*  
*Use case* digunakan untuk menggambarkan sebuah jalinan hubungan atau relasi pada aktor dengan sistem informasi yang akan dibuat.
2. *Class Diagram*  
*Class diagram* atau diagram kelas merupakan tipe diagram yang paling banyak dipakai, digunakan untuk menggambarkan struktur pada sistem yaitu pendefinisian kelas seperti atribut, operasi, dan hubungan antara kelas yang akan dibuat untuk membentuk arsitektur sistem yang dibuat.
3. *Sequence Diagram*  
*Sequence diagram* adalah diagram yang digunakan untuk menjelaskan interaksi pada objek-objek yang terlibat di dalam *use case* yang disusun berdasarkan urutan waktu. Dengan kata lain, *Sequence diagram* digunakan untuk mengetahui gambaran yang ada pada sistem secara tahap demi tahap.

#### 4. Activity Diagram

Diagram aktivitas berguna untuk menjelaskan tentang urutan kegiatan (aliran kerja) sistem. Diagram aktivitas biasanya digunakan untuk menggambarkan kegiatan, objek, *state*, transisi *state* dan *event*.

### 2.5 Metode Waterfall

Metode *Waterfall* adalah suatu pengembangan perangkat lunak yang dilakukan berurutan, untuk model pengembangannya dapat dianalogikan seperti kemajuan dipandang sebagai terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian [6]. Metode *Waterfall* memiliki beberapa tahapan dalam pengembangannya yaitu sebagai berikut.

#### 1. Analisa

Dalam tahapan ini merupakan mempersiapkan dan menganalisa terhadap kebutuhan sistem. Pengumpulan data-data yang dibutuhkan dalam pembuatan sistem web agar dapat merancang konsep serta antarmuka web.

#### 2. Desain

Proses perancangan desain ini difokuskan pada: struktur data, arsitektur perangkat lunak, representasi interface, dan detail (algoritma) *procedural* interaksi antar objek.

#### 3. Penulisan kode program

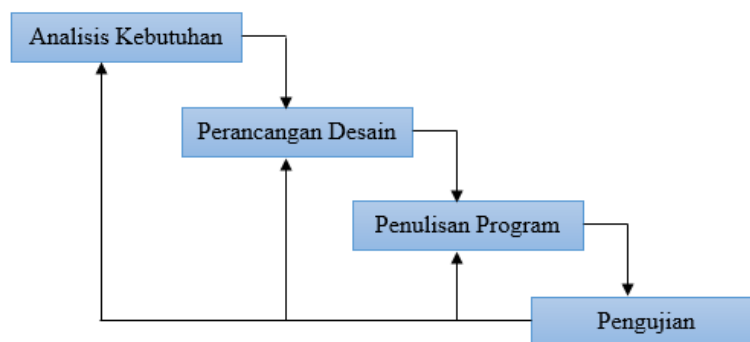
Pada tahap ini, merupakan tahapan secara nyata dalam mengerjakan suatu sistem yang lebih berfokus pada hal teknis. Perancangan perangkat lunak direalisasikan sebagai serangkaian program.

#### 4. Testing

Tahapan ini adalah tahapan akhir dalam pembuatan sistem unit melibatkan verifikasi bahwa unit program telah memenuhi syarat atau spesifikasi tahapan analisa sistem, perancangan *design*, dan penulisan program.

## 3 Metodologi Penelitian

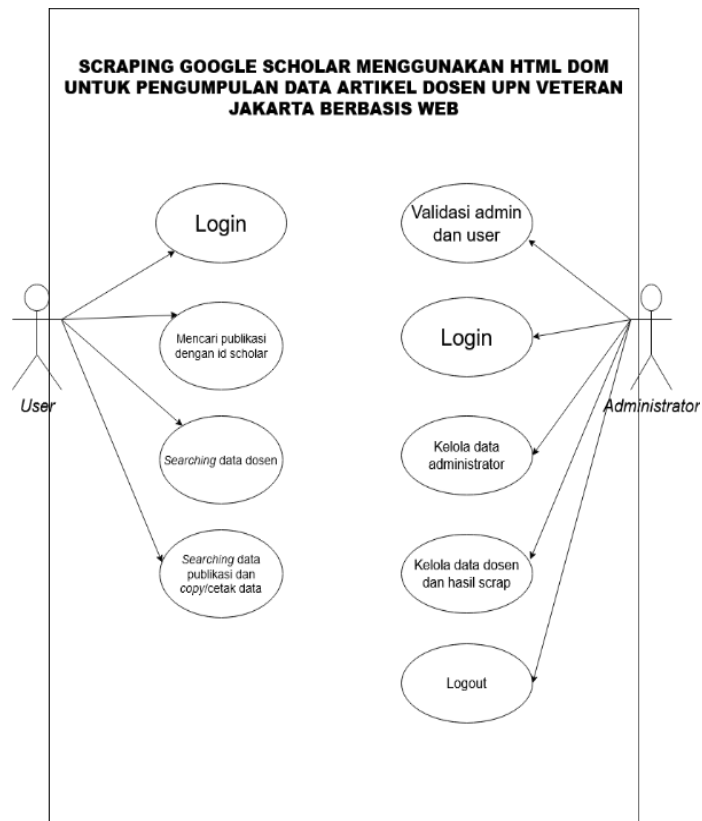
Proses pengerjaan *website* dimulai dari pengumpulan id dosen pada *Google Scholar*, lalu merancang kode program untuk *scraping* data artikel dengan bantuan PHP Simple HTML DOM Parser [7]. Dalam pengembangan sistem digunakan *waterfall model* yang melibatkan 4 fase sebagai berikut :



Gambar. 2. Tahap pengembangan sistem web.

### 3.1 Analisis Kebutuhan Sistem

*Website* berisi data publikasi dosen UPN Veteran Jakarta dibuat dengan bahasa pemrograman PHP, dan *database* phpMyAdmin yang berbentuk MySQL. Tahap awal dalam pembuatan web adalah melakukan analisa keperluan sistem, salah satunya dengan mengumpulkan *id scholar* para dosen UPN Veteran Jakarta yang digunakan untuk melakukan penarikan data artikel dosen. Dalam *website* berisi 2 *role* yaitu *admin* dan *user* (pengguna) yang memiliki fungsi yang berbeda.



**Gambar. 3.** Use Case Diagram

Gambar 3 berisi uraian kegiatan yang dapat dilihat bahwa *admin* dapat mengelola akun para pengguna web, *scraping*, dan melakukan edit. Sedangkan *user* hanya dapat melihat data dosen tanpa pengeditan serta dapat melakukan *copy* dan cetak data.

### 3.2 Perancangan Database

Database diberi nama db\_webscrap yang berisi 4 tabel yaitu:

1. Tabel anggota (tbl\_anggota)

Tabel anggota berisi nama-nama dosen yang di-*input* oleh *admin*. Jika tidak diketahui *nidn*, maka dapat mengisi dengan nilai 0 atau angka lainnya. Hal ini tidak mengganggu proses *scraping*, sebab *scraping* hanya membutuhkan *id scholar* dosen. Berikut struktur tabel anggota :

**Tabel 1.** Struktur Tabel Anggota

No	Nama Field	Type	Keterangan
1	id	int(11)	Id dosen ( <i>key increment</i> )
2	nidn	varchar(20)	Nomor NIDN dosen
3	nama	varchar(250)	Nama dosen
4	email	varchar(50)	Email dosen
5	prodi	varchar(30)	Prodi dosen
6	id_fakultas	varchar(35)	Id fakultas
7	id_scholar	varchar(15)	Id <i>Google Scholar</i> dosen
8	link	varchar(100)	link dari akun <i>Google Scholar</i> dosen

2. Tabel Fakultas (tbl\_fakultas)

Tabel fakultas berisi nama-nama fakultas serta jumlah profil dan jurnal yang didapatkan dari jumlah profil dan jurnal yang ada di tabel anggota. Berikut struktur tabel fakultas :

**Tabel 2.** Struktur Tabel Fakultas

No	Nama Field	Type	Keterangan
1	id	int(5)	Id fakultas ( <i>key increment</i> )
2	fakultas	varchar(45)	Nama fakultas
3	jum_profile	int(10)	Jumlah profil tiap fakultas
4	jum_jurnal	int(10)	Jumlah jurnal tiap fakultas

3. Tabel Hasil Scraping (tbl\_scrap)

Tabel ini berisi hasil *scrap Google Scholar* yang dilakukan dengan *id\_scholar*. Berikut struktur pada tabel hasil *scraping* :

**Tabel 3.** Struktur Tabel Hasil Scraping

No	Nama Field	Type	Keterangan
1	judul	text	Judul karya ilmiah (paper)
2	id_scholar	varchar(50)	Id <i>Google Scholar</i> dosen
3	pengarang	varchar(50)	Nama-nama pengarang paper
4	tahun_terbit	varchar(50)	Tahun diterbitkannya paper
5	jurnal	text	Nama jurnal publikasi
6	citation	int(5)	Jumlah sitasi paper

4. Tabel User (tbl\_user)

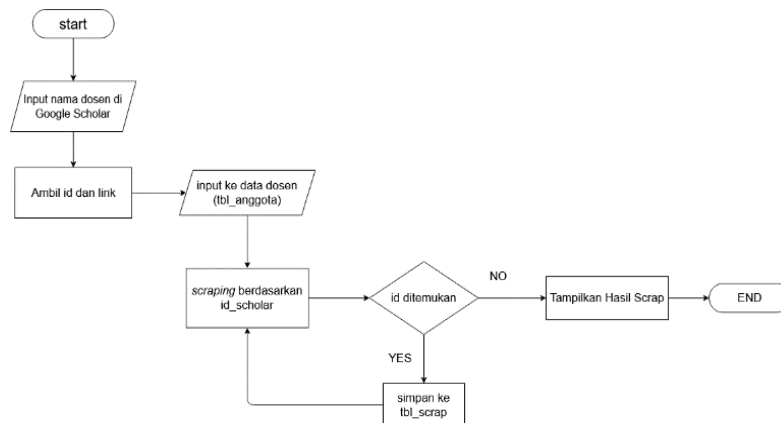
Tabel ini berisikan data *user* yang di-input oleh *admin* dalam pembuatan akun *website*. Berikut struktur tabel *user* :

**Tabel 4.** Struktur Tabel User

No	Nama Field	Type	Keterangan
1	id	int(3)	Id akun ( <i>key increment</i> )
2	nama	varchar(200)	Nama panjang <i>user</i>
3	username	varchar(50)	<i>Username</i> akun
4	password	varchar(200)	Kata sandi akun
5	email	varchar(100)	Email <i>user</i>
6	foto	varchar(50)	Foto <i>user</i>
7	level	enum('admin', 'user')	Memberikan role <i>admin</i> atau <i>user</i>

### 3.3 Penyusunan Kode Web

Tahapan ini, rancangan yang telah dibuat diterjemahkan ke dalam bentuk bahasa pemrograman PHP [8]. *Coding* berfungsi menjalankan aplikasi dengan logika algoritma yang diterjemahkan. Agar aplikasi dapat berjalan sesuai tujuan yang diinginkan, maka *coding* harus sesuai *flowchart* alur rancangan sebagai berikut :



Gambar. 4. Alur rancangan

## 4 Hasil dan Pembahasan

### 4.1 Tampilan web

#### 1. Halaman *Log in*

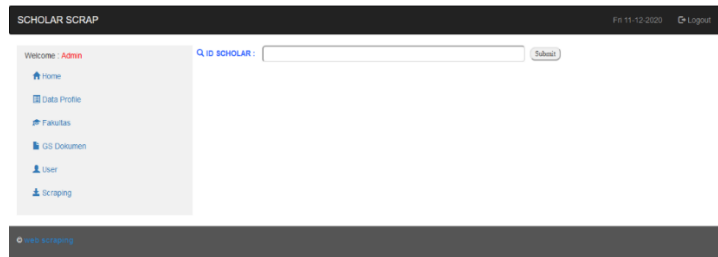
Pada halaman *login*, pengguna diwajibkan memasukkan *username* dan *password* akun. Bagi pengguna yang belum memiliki akun, maka diarahkan untuk menghubungi administrator untuk mendaftarkan akunnya. Jika proses *login* berhasil, maka akan muncul notifikasi *login* sesuai *role* pada akun. Tampilan halaman *login* dapat dilihat pada gambar 5.

The screenshot shows a 'Sign In' form with a light blue header. It contains two input fields: 'username' with a person icon and 'password' with a lock icon. Below the fields is a blue 'Login' button. Underneath the button, there is a 'Contact Us' section with an 'Attention' message: 'Daftar Sebagai Anggota Langsung dengan Admin! Setelah daftar akan diberikan hak Akses Akun.' and a contact number 'Admin : +62823.....' with a blue button.

Gambar. 5. Tampilan halaman *login*

#### 2. Halaman awal

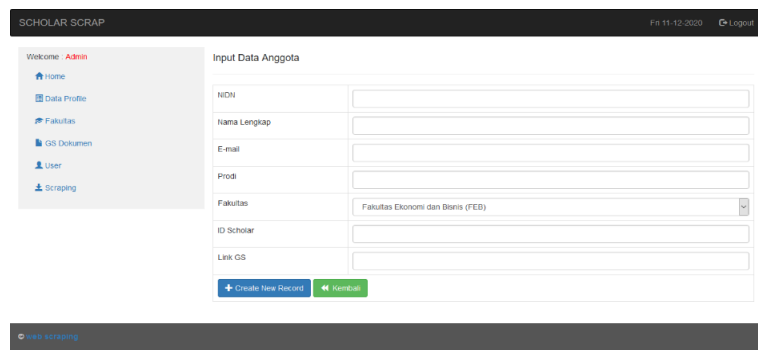
Saat web dijalankan, akan ditampilkan halaman awal yang berisi *search box* untuk mencari paper dengan *id scholar*. Berikut tampilan halaman awal pada *role admin* :



**Gambar. 6.** Tampilan halaman awal (beranda) pada role *admin*

3. Menu *Data Profile*

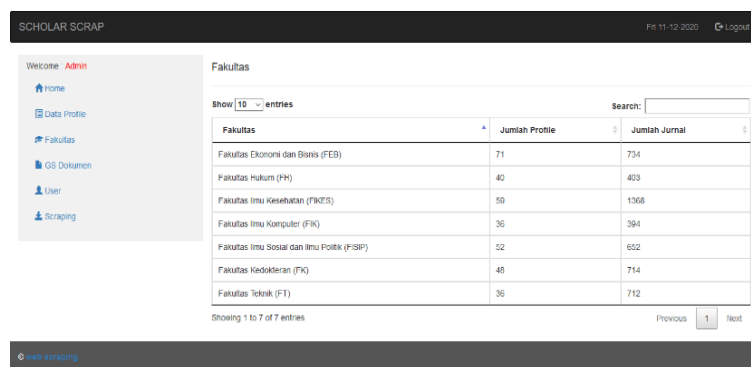
Menu *data profile* berisi data dosen yang telah di-*input*. Proses *input* hanya dapat dilakukan oleh *admin*. Hasil *input* akan dimasukkan ke dalam tabel *tbl\_anggota*. Role *user* hanya dapat mencari data dosen berdasarkan fakultas, dan mengurutkan baik dari nama, dan prodi berdasarkan urutan abjad. Berikut tampilan saat *input* data anggota :



**Gambar. 7.** Tampilan *input* data anggota

4. Menu Fakultas

Halaman menu fakultasi berisi jumlah profil dan jurnal berdasarkan fakultas seperti yang terlihat pada gambar 8. Tampilan menu fakultas baik pada *role admin* maupun *user* sama. Menu fakultas merupakan hasil dari tampilan pada *tbl\_fakultas*, di mana data dari tabel tersebut bersifat otomatis mengikut jumlah profil atau jurnal yang ada di *tbl\_anggota*.



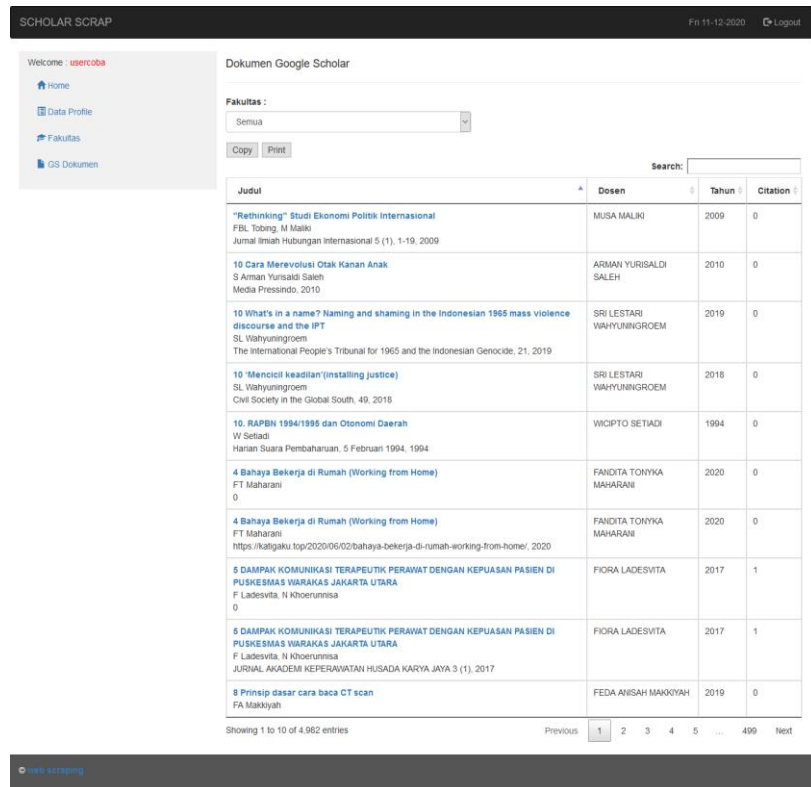
**Gambar. 8.** Tampilan halaman menu Fakultas

5. Menu *GS Document*

Menu ini berisi hasil dari proses *scraping* yang dilakukan pada menu *scraping* berdasarkan data dari tabel *tbl\_anggota*. Proses *scraping* dilakukan dengan *id scholar* pada *tbl\_anggota*. Hasil dari proses *scraping* akan disimpan pada *tbl\_scrap* yang akan ditampilkan pada menu *GS Document*. Hasil *scrap* dapat kita *copy* ke



dalam *data frame* (misal Excel) dan print ke bentuk pdf. Berikut tampilan halaman menu *GS Document* pada role *user* :

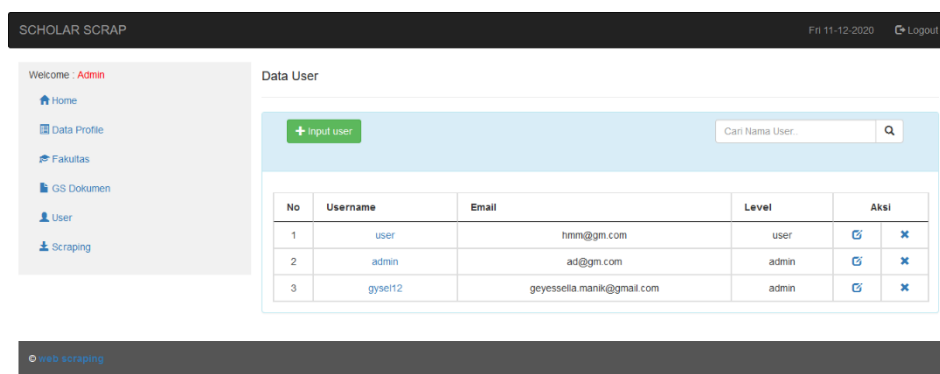


**Gambar. 9.** Halaman menu *GS Document* role *user*

Tampilan menu *GS Document* baik dari *role admin* atau *user* sama. *User* dapat mengurutkan judul, dosen, tahun, dan sitasi berdasarkan urutan abjad atau angka terkecil. *User* juga dapat menekan judul paper yang akan diarahkan ke pencarian judul di *Google Scholar*.

#### 6. Menu *User*

Menu ini berisi data administrator atau validasi *role* pada akun. Menu ini hanya terdapat pada *role administrator*. Untuk mendaftar atau membuat akun, *admin* harus mengisi data-data yang dibutuhkan yang akan disimpan ke dalam tabel *tbl\_user*. Selain itu, *admin* juga dapat melakukan search nama *user* pada *search box* yang tersedia. Berikut tampilan pada menu *User* :



**Gambar. 10.** Halaman menu *User*

## 7. Menu Scraping

Halaman berisi konfirmasi untuk melakukan *scraping* dan hanya terdapat pada *role admin*. Selain untuk melakukan konfirmasi *scraping*, menu ini juga dimaksud agar *admin* dapat memeriksa kembali data seperti *id scholar* untuk dilakukan proses *scraping*. Berikut tampilan menu *Scraping* :

Welcome : Admin

Lakukan scraping data dari tbl\_anggota

No	NIDN	Nama	Prodi	ID Scholar
1	0001039203	SANTIKA SARI	S1 TEKNIK INDUSTRI	DgbZaaIAAAAJ
2	0002058205	FAJAR RAHAYU IKHWANUL	S1 TEKNIK ELEKTRO	xav_anAAAAAJ
3	0002078509	FITRI WAHYUNI	S1 TEKNIK MESIN	L-9PU_oAAAAJ
4	0006065612	BUDHI MARTANA	S1 TEKNIK MESIN	0lqeICIAAAAJ
5	0009077102	LUH KRISNAWATI	S1 TEKNIK INDUSTRI	MJEpBgAAAAAJ
6	0010016701	SITI ROHANA NASUTION	S1 TEKNIK INDUSTRI	nCEntf6gAAAAAJ
7	0010097407	WIWIN SULISTYAWATI	S1 TEKNIK PERKAPALAN	yWaut9wAAAAAJ
8	0014119301	AKALILY MARDHYA	S1 TEKNIK ELEKTRO	AfoM6eSAAAAAJ
9	0017128904	ALINA CYNTHIA DEWI	S1 TEKNIK INDUSTRI	CSF70AEAAAAAJ
10	0018078606	JAMES JULIAN	S1 TEKNIK MESIN	r_avr7wIAAAAJ
11	0018078903	AKHMAD NIDHOMUZZAMAN	S1 TEKNIK INDUSTRI	rGmC9LUAAAAAJ
12	0018129202	MOHAMMAD RACHMAN WALUYO	S1 TEKNIK INDUSTRI	Zxb5h6gAAAAAJ
13	0020046008	LILIK ZULAHAH	S1 TEKNIK INDUSTRI	yEC329wAAAAAJ
14	0020075807	MOHAMMAD RUSDY HATUWE	S1 TEKNIK PERKAPALAN	8okDbk4AAAAAJ
15	0020108801	MUHAMMAD ARIFUDIN LUKMANA	S1 TEKNIK MESIN	o7YDW5kAAAAAJ
16	0021108304	NOVERDO SAPUTRA	S1 TEKNIK MESIN	Q2oN94IAAAAAAJ
17	0021115709	HALIM MAHFUDH	S1 TEKNIK INDUSTRI	C_oY2KAAAAAJ
18	0022026106	MUHAMAD AS'ADI	S1 TEKNIK INDUSTRI	Qk7Z8YEAAAAAJ
19	0024026212	ISWADI NUR	S1 TEKNIK PERKAPALAN	g2mcs0AAAAAJ
20	0025038306	TATIK JUWARIYAH	S1 TEKNIK PERKAPALAN	69g3tP8AAAAAJ

Jumlah : 352 Anggota

← Sisron → tbl\_scrap

Gambar.11. Halaman menu *Scraping*

## 4.2 Pengujian (Testing)

Pada tahap ini, *website* yang dirancang akan dilakukan *testing* guna mencari kesalahan kode dan kesalahan logika. Hasil dari *testing* yaitu proses *scraping* dan penyimpanan ke *database* berjalan baik. Namun, ditemukan bahwa *id scholar* dapat berubah-ubah dan proses *scraping* yang sangat lama yaitu  $\pm 30$  menit untuk 352 profil. *Website* dapat menampilkan hasil sitasi setiap paper dengan baik, namun tidak dapat menampilkan sitasi berdasarkan per tahun, sehingga sitasi yang tertampil adalah sitasi keseluruhan tahun.

Diperlukan pemeriksaan berkala *id scholar* dosen, terlihat pada *id scholar* pada data terakhir *tbl\_scrap* apakah sama dengan *id scholar* terakhir pada *tbl\_anggota* atau tidak. Dalam melakukan pencarian di *search box* pada menu *Data Profile*, masih terdapat kegagalan pencarian nama jika tidak memilih fakultas terlebih dahulu. Dalam menampilkan *tbl\_scrap* pada menu *GS Document* masih berlangsung lama. Hal ini dikarenakan banyaknya *query* pada *database* sehingga membutuhkan proses loading dengan durasi cukup lama.

## 5 Kesimpulan dan Saran

### 5.1 Kesimpulan

1. *Website* berhasil melakukan *scraping Google Scholar* sehingga mendapatkan artikel jurnal dosen UPN Veteran Jakarta.
2. Web dapat menyimpan data hasil *scraping* pada *database*.
3. Dengan adanya aplikasi ini, mempermudah *user* dalam mengumpulkan, dan rekapitulasi mengenai informasi artikel/jurnal ilmiah dosen UPN Veteran Jakarta.

### 5.2 Saran

*Website* perlu dikembangkan lebih lanjut untuk penyempurnaan baik fitur-fitur yang ditampilkan atau tampilan dari *website*. Seperti menu *scraping* dapat ditambahkan *checklist* atau *check box* untuk memilih nama-nama dosen yang akan dilakukan *scraping* atau tidak, sehingga tidak perlu melakukan pengulangan *scraping*. Kemudian dibutuhkan perbaikan dalam menampilkan sitasi sehingga sitasi yang ditampilkan dapat dibedakan berdasarkan setiap tahunnya.

## Referensi

- [1] Google Cendekia. (2020, September 26). Diakses pada Maret 26, 2021, dari Wikipedia: [https://id.wikipedia.org/wiki/Google\\_Cendekia](https://id.wikipedia.org/wiki/Google_Cendekia).
- [2] Triyono. (2020, April 30). Pengertian dan Manfaat Google Scholar [Halaman web]. Diakses dari <https://raharja.ac.id/2020/04/30/pengertian-dan-manfaat-google-scholar>.
- [3] Ayani, D. D., Partiwi, H. S., Muhardi, H. (2019). Implementasi *Web Scraping* untuk Pengambilan Data pada Situs Marketplace. *Jurnal Sistem dan Teknologi Informasi*, 7(4), 257-262.
- [4] Mitra, V., Sujaini, H., Negara, A. B. P. (2017). Rancang Bangun Aplikasi Web Scraping untuk Korpus Paralel Indonesia-Inggris dengan Metode HTML DOM. *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, 5(1), 36-41.
- [5] Dimmay, H. M. (2016). Pengertian Database MySQL dan phpMyAdmin [Halaman web]. Diakses dari <http://hadribonjay.it.student.pens.ac.id/>.
- [6] Alfina, O., Harahap, F. (2019). Pemodelan UML Sistem Pendukung Keputusan dalam Penentuan Kelas Siswa Tunagrahita. *Methomika: Jurnal Manajemen Informatika & Komputerisasi Akuntansi*, 3(2), 143-150.
- [7] Indiyah, F. H. (2016). Pengembangan Sistem Manajemen Surat Menyurat (Masmatik) Jurusan Matematika. *KNPMP I*, 906-914.
- [8] Rahmatulloh, A., & Gunawan, R. (2020). *Web Scraping with HTML DOM Method for Data Collection of Scientific Articles from Google Scholar*. *Indonesian Journal of Information Systems*, 2(2), 95-104.
- [9] Komang. (2019). Web Crawler Sederhana dengan PHP Simple HTML DOM Parser [Halaman web]. Diakses dari <http://komang.my.id/2019/05/14/web-crawler-sederhana-dengan-php-simple-html-dom-parser/>.