WEBSITE PENELUSURAN ARTIKEL ILMIAH DENGAN MEMANFAATKAN PARSCIT, GOOGLE SCHOLAR DAN MENDELEY API

Indra Ruslan[1], Adi Wibowo[2], Resmana Lim[3]

Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) - 8417658
E-mail:expeliarnum@gmail.com[1], adiw@petra.ac.id[2], resmana@petra.ac.id[3]

ABSTRAK

Jurnal ilmiah berperan penting sebagai referensi bagi banyak orang ketika melakukan sebuah penelitian. Hal ini disebabkan karena jurnal ilmiah memaparkan suatu pembahasan secara ilmiah yang dilakukan oleh penulis atau peneliti untuk membagikan suatu hal secara logis dan sistematis kepada para pembacanya. Banyak perusahaan besar seperti Google, Mendeley, Endnote, Refworks, Zotero membuat sebuah *website* untuk menampung semua jurnal yang dipublikasi. Namun setiap *website* memiliki koleksi dan kapasitas jurnal yang berbeda-beda.

Oleh karena itu, pada proyek ini dibuat sebuah *website* yang berguna untuk mencari koleksi jurnal yang sesuai dengan kebutuhan. Proyek ini dibuat untuk mengefisienkan pencarian jurnal ilmiah yang ada di Mendeley dan Google Scholar dengan memanfaatkan data *paper* hasil ekstraksi sitasi ParsCit. Aplikasi *website* ini dibuat menggunakan *Hypertext Preprocessor* (PHP) sebagai bahasa pemrograman dan MySQL sebagai *database server*.

Dari hasil pengujian, dapat diketahui bahwa proses pencarian *paper* di Mendeley mendapatkan hasil yang lebih banyak dibandingkan dengan proses pencarian *paper* di Google Scholar. Hal ini disebabkan karena Mendeley menyediakan API untuk akses data *paper*, sedangkan Google Scholar tidak.

Kata kunci: API, *Document Object Model*, Google Scholar, Mendeley, Mendeley API, ParsCit, *Similarity.*

ABSTRACT

Scientific papers are important and crucial for many people during a research. This can be affected by usefulness of scientific paper which explain study scholarly and be made by researcher to share things legitimate and systematic to the readers. There are a lot of big company like Google, Mendeley, Endnote, Redworks, Zotero which made a website to collect a lot of scientific papers which have been published. But unfortunately, every website has different collection and capacity of papers.

In this project, a useful website application is built to search papers collection that needed by website's users and members. This project made to explore scientific papers from Mendeley and Google Scholar using the result from ParsCit extraction. This website application is made using Hypertext Preprocessor (PHP) as the programming language and MySQL as the database server.

From the results of system implementation and testing of this website, can be concluded that from search process on Mendeley, we can get more results than from search process on Google Scholar. That can be happened because Mendeley provided API for programmer to access paper data on Mendeley's database and Google Scholar don't provide API.

***Keywords**:* API, *Document Object Model*, Google Scholar, Mendeley, Mendeley API, ParsCit, *Similarity.*

1. PENDAHULUAN

Pengguna internet dari tahun ke tahun semakin meningkat tajam, selain itu jumlah publikasi data yang diunduh maupun diunggah di internet mencapai jutaan data setiap harinya. Hal ini disebabkan karena penggunaan internet yang meningkat hampir di setiap bidang yang ada.

Salah satu contohnya adalah di bidang kesehatan, secara umum satu universitas di Amerika bisa menghasilkan lebih dari 1.000 jurnal per tahun, apabila terdapat 1.000 universitas di seluruh dunia, maka total jurnal yang terbit setiap tahunnya mencapai 1.000.000 jurnal baru [1].

Banyak perusahaan besar seperti Google, Mendeley, Endnote, Refworks, Zotero membuat sebuah *website* untuk menampung semua jurnal yang dipublikasi. Akan tetapi setiap *website* tidak mempunyai kapasitas dan koleksi jurnal yang sama, sehingga untuk mendapatkan jurnal yang sesuai dengan kebutuhan, perlu dilakukan pencarian di banyak *website* sehingga diperlukan banyak *account* untuk dapat mengakses jurnal yang dicari.

Lalu bagaimana caranya mendapat jurnal yang sesuai dengan kebutuhan, tanpa harus melakukan *searching* ke *website* yang lain? Pastinya, ada sebuah *website* baru yang terhubung dengan *website* yang menampung jurnal ilmiah.

Cara kerja sistem *website* adalah sebagai berikut, pada awalnya, atribut-atribut jurnal maupun *paper* yang di-*upload* oleh *member* akan diambil dengan menggunakan ParsCit. ParsCit melakukan ekstraksi sitasi terhadap atribut *paper* sehingga didapatkan judul, nama penulis, afiliasi, email, abstrak, kata kunci dan daftar referensi dari *input paper*.

Selanjutnya dilakukan pencarian di Google Scholar dan Mendeley untuk menemukan *paper* yang berhubungan dengan *input paper* yang dimasukkan dengan parameter atribut *paper* yang didapatkan dari hasil ParsCit.

2. TINJAUAN PUSTAKA

2.1 Mendeley API

Mendeley API menggunakan OAuth untuk otentikasi. Para pengembang sistem harus mendaftarkan aplikasi dan mendapatkan *consumer key* dan *consumer secret*. Seluruh metode spesifik dari pengguna membutuhkan otentikasi 3Leg OAuth dan membutuhkan aplikasi pengguna yang memperbolehkan akses terhadap data. Seluruh *request* yang diminta terhadap Mendeley API harus merupakan GET *requests*.

2.2 Google Scholar API

Berbeda dengan API pada *website* Mendeley, Google Scholar tidak memiliki API. Oleh sebab itu, ada beberapa sumber yang mencoba untuk menelusuri API yang terdapat pada Google Scholar, salah satunya yakni dengan menggunakan Python. Sebagai hasilnya, akan didapatkan *format dictionary* yang baik dengan *field addressed* menggunakan *key* sebagai hasil dari kode tersebut.

Untuk menggunakan Google Scholar API, ada syarat dan aturan yang perlu diperhatikan, salah satunya adalah Google Scholar API tidak memerlukan *request authentication*, selain itu parameter yang harus dibutuhkan adalah sebuah *terms* dan dapat digunakan *generic search* untuk membuat hasil pencarian semakin spesifik.

2.3 ParsCit

ParsCit dikembangkan oleh *Information Science and Technology* (IST) *Penn State University* dan *National University of Singapore* (NUS). ParsCit terbukti baik dan telah diadopsi oleh beberapa sistem lainnya misalnya Mendeley.com. *Software* ini dibuat oleh Min-Yen Kan, Isaac G. Councill, C. Lee Giles, dan Minh-Thang Luong pada tahun 2008 [2]. *Software* ini bersifat *Open Source* tetapi dilindungi oleh GNU *Lesser General Public License* yang dipublikasikan oleh *Free Software Foundation*.

Isi di dalam *software* ini yakni dapat dilakukan modifikasi pada kode programnya dan boleh didistribusikan ulang selama masih memenuhi kondisi dalam *license* tersebut [3]. *Software* ini digunakan untuk mengolah sebuah *paper* atau *journal* elektronik dengan tujuan untuk mengambil informasi-informasi penting secara otomatis dari *journal* atau *paper* seperti, judul, penulis, institusi, alamat, tahun, email, abstrak, kata kunci, dan sitasi (daftar referensi) [4].

2.4 Metode *Parsing*

Parsing dilakukan untuk memudahkan proses mendapatkan *metadata* yang dibutuhkan oleh pengembang program. Metode *Parsing* yang digunakan yakni menggunakan DOM (*Document Object Model*).

Document Object Model merupakan sebuah API (*Application Programming Interface*) untuk HTML yang valid dan dokumen-dokumen bertipe XML, berarti struktur *logical* dari dokumen-dokumen dapat diakses dan dimanipulasi [5]. Di dalam spesifikasi DOM, XML digunakan sebagai sebuah cara untuk mewakili beberapa jenis informasi yang dapat disimpan di dalam sistem yang berbeda dan kebanyakan dilihat sebagai data dibandingkan sebagai dokumen. Namun, XML menampilkan data sebagai dokumen dan DOM digunakan untuk mengelola data tersebut.

Teknik untuk *parsing* data yang digunakan adalah DOM (*Document Object Model*) dengan mempertimbangkan beberapa hal yang dianggap menguntungkan apabila menggunakan DOM yakni kecepatan dalam melakukan *parsing*, kemudahan dalam pengembangan program dan kualitas hasil *parsing*.

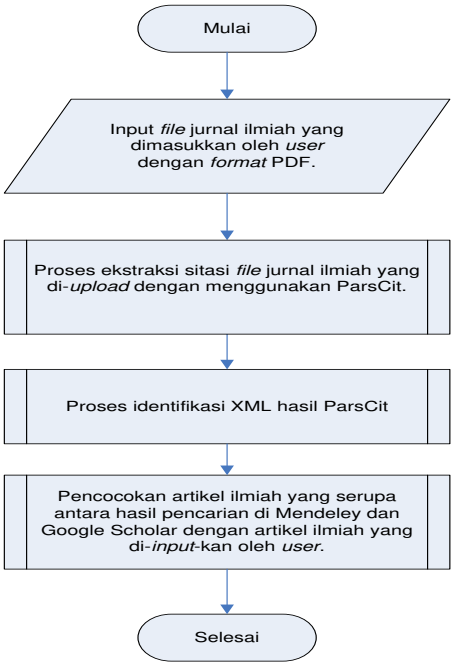
3. DESAIN SISTEM

3.1 Garis besar sistem kerja *website* penelusuran artikel ilmiah.

File jurnal ilmiah yang di-*upload* oleh *user* adalah jurnal ilmiah dengan *format* PDF. Kemudian jurnal ilmiah yang dimasukkan akan diekstraksi sitasi oleh sistem dengan menggunakan ParsCit. Proses ini bertujuan untuk mendapatkan metadata dari jurnal dan metadata dari jurnal-jurnal yang menjadi referensi dari jurnal yang di-*input*-kan yang kemudian dimanfaatkan untuk proses pencarian jurnal serupa di Mendeley maupun Google Scholar.

Proses selanjutnya yaitu proses *download* XML hasil ParsCit. Setelah mendapatkan XML, maka dilakukan identifikasi terhadap XML yang didapatkan. Proses ini bertujuan untuk mengambil metadata dari Google Scholar dan Mendeley yang selanjutnya akan dicocokkan dengan metadata dari jurnal yang dimasukkan. Pada proses pengambilan metadata dari Mendeley, *website* akan melakukan *request* terhadap Mendeley API dan Mendeley API akan merespons dengan menampilkan jurnal-jurnal ilmiah yang berelasi dengan jurnal ilmiah yang dimasukkan. Sedangkan pada Google Scholar, dilakukan *parsing* untuk mendapatkan metadata yang dibutuhkan, karena Google Scholar tidak menyediakan API.

Proses yang terakhir yaitu pencocokan artikel ilmiah yang serupa antara Mendeley dan Google Scholar dengan jurnal ilmiah yang dimasukkan dengan menggunakan metode *similarity*.



Gambar 1. Diagram alir garis besar sistem kerja *website* penelusuran artikel ilmiah.

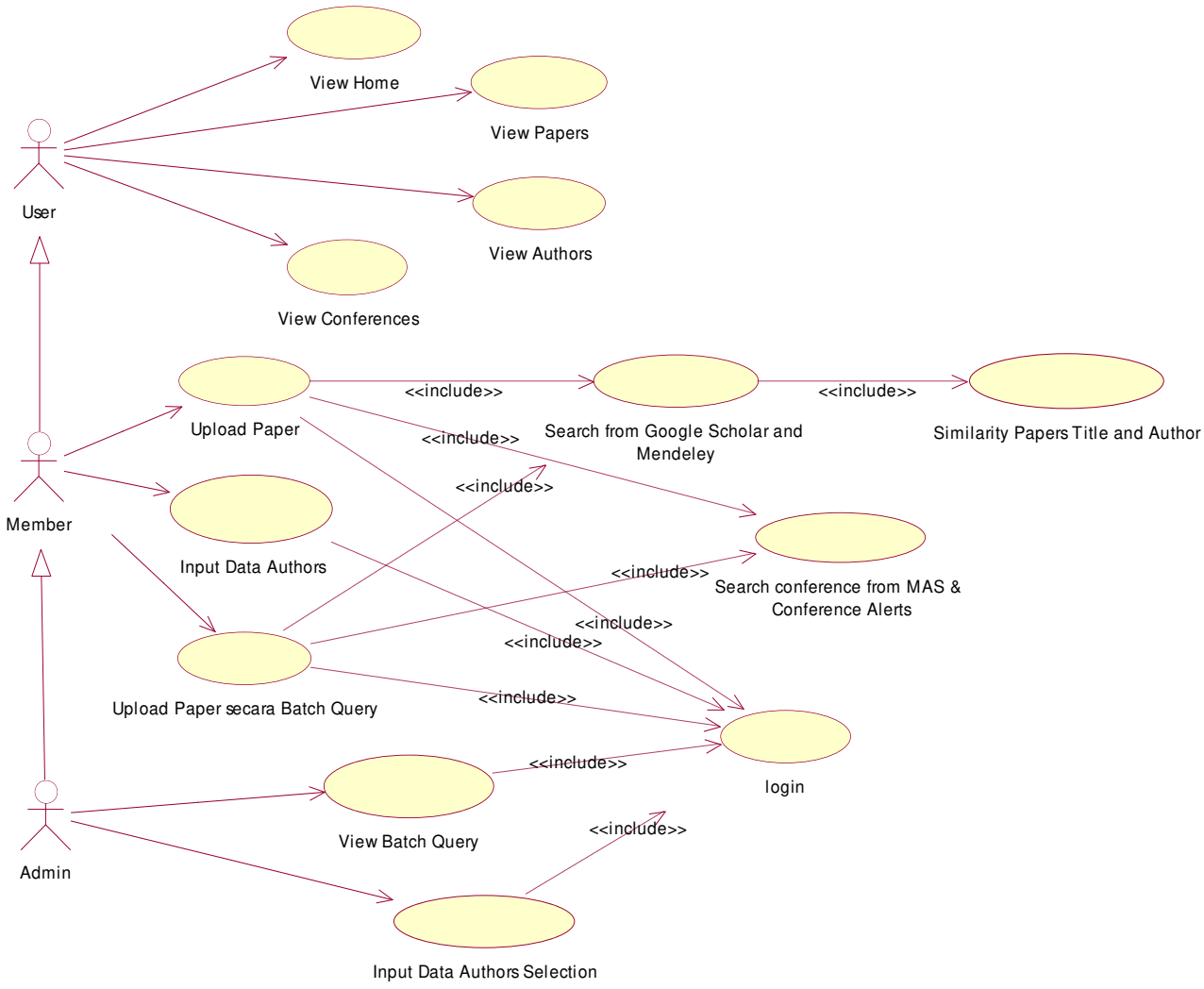
3.2 *Use Case Diagram*

User dapat melihat halaman *home, papers, authors* dan *conferences*. Apabila *user* sudah mendaftar menjadi *member* melalui *menu sign up*, maka *user* tersebut menjadi *member* yang memiliki hak akses lebih daripada *user* biasa. *Member* memiliki hak akses yang lebih banyak dibandingkan dengan *user*.

Member dapat melakukan seluruh aktivitas yang dapat dilakukan oleh *user*, ditambah fitur memasukkan *paper*, memasukkan maupun melengkapi data pengarang jurnal dan memasukkan *paper* secara *batch query*. Apabila *member* melakukan *upload paper* secara *batch*, sistem akan melakukan pencarian *paper* secara *batch* sehingga *member* tidak perlu menunggu sistem untuk melakukan pencarian.

Sedangkan *admin* bertugas untuk mengawasi seluruh aktivitas yang terdapat di dalam sistem *website*. Orang yang memiliki hak akses sebagai *admin* dapat melihat proses *batch* dari *paper* yang telah di-*upload* oleh *member* dan statusnya, *admin* juga dapat menganalisa dan menyeleksi data pengarang jurnal yang dimasukkan oleh *member*.

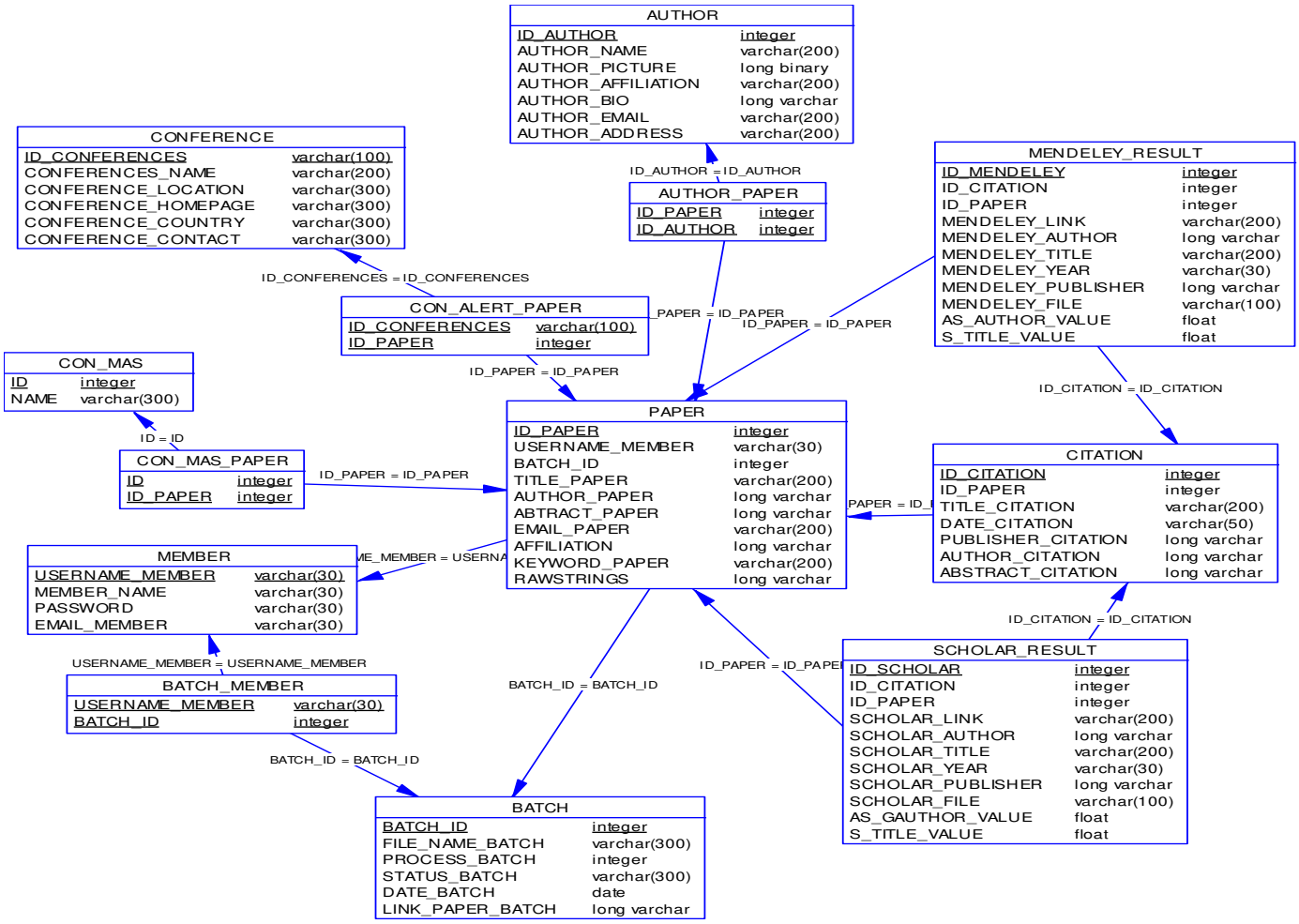
Untuk proses *upload paper* dan *upload paper* secara *batch query*, akan dilakukan pula proses pencarian *paper* referensi di Google Scholar dan Mendeley. Setelah proses pencarian *paper* referensi di Google Scholar dan Mendeley selesai, dilanjutkan dengan perhitungan nilai *similarity* (kemiripan) dari *paper-paper* yang ditemukan dengan parameter judul (*title*) *paper* dan penulis (*author*) *paper*. Selain itu juga terdapat proses pencarian konferensi yang berhubungan dengan *paper* di Microsoft Academic Search dan Conference Alerts apabila *paper* yang di-*upload* memiliki kata kunci (*keyword*) *Use case diagram* dari *website* dapat dilihat pada Gambar 2.



Gambar 2. *Use case diagram* dari *user*, *member* dan *admin*.

3.3 *Entity Relationship Diagram* (ERD)

Untuk penyimpanan data di dalam *database*, maka dibuat *Entity Relationship Diagram* (ERD) agar memudahkan dalam hal penyusunan atribut pada masing-masing *entity* dan mencegah terjadinya redundansi data. ERD yang dibuat terbagi menjadi *Conceptual* ERD dan *Physical* ERD (Dapat dilihat pada Gambar 3).



Gambar 3. *Physical Entity Relationship Diagram*.

4. IMPLEMENTASI

Berikut akan dijelaskan mengenai implementasi *website* dalam potongan / segmen *source code* dalam bahasa pemrograman PHP, MySQL dan HTML. Contoh proses *parsing* dapat dilihat pada Segmen Program 1 dan Segmen Program 2.

```
curl_setopt($curl,CURLOPT_URL,$gbibtex);
curl_setopt($curl,CURLOPT_RETURNTRANSFER,1);
curl_setopt($curl,CURLOPT_COOKIE,"GSP=ID=451a3558f
27052ed:CF=4:S=pYR5xjParrDcP7Gx");

        $respbib = curl_exec($curl);
        $poshrefb1 = strpos($respbib,"title={");
        $poshrefb2      =      strpos($respbib,"}",
        $poshrefb1+7);
        if($poshrefb1 !== false && $poshrefb2 !==
        false)
        {
                $bibtitle =
                substr($respbib,$poshrefb1+7,($posh
                refb2-($poshrefb1+7)));
        }
```

Segmen Program 1. *Source code* untuk mengambil atribut judul *paper* (*title*) di dalam *bibtex* dari *paper* yang didapatkan dari hasil pencarian di Google Scholar..

Deklarasi awal yakni melakukan *set* terhadap opsi cURL menggunakan fungsi curl_setopt() pada variabel $gbibtex. Kemudian eksekusi cURL menggunakan fungsi curl_exec().

Selanjutnya dilakukan *parsing* untuk memotong dan mengambil nilai judul paper (*title*) yang terdapat di dalam title={ yang dimasukkan ke dalam variabel $poshrefb1 sampai dengan } yang dimasukkan ke dalam variabel $poshrefb2 dengan menggunakan fungsi strpos(). Apabila $poshrefb1 dan $poshrefb2 bernilai *true* (ada), maka *title* diambil dari akhir variabel $poshrefb1 sampai awal variabel $poshrefb2 menggunakan fungsi substr() dan selanjutnya nilai *title* dimasukkan ke dalam variabel $bibtitle.

```
$link  =  $mendeleylink.urlencode($properties)."/?
consumer_key= $con sumer_key&items=20";
        curl_setopt($curl,CURLOPT_URL,$link);
        curl_setopt($curl,CURLOPT_RETURNTRANSFER,1
        );
        $resp = curl_exec($curl);
        $responsem = json_decode($resp);
        if(isset($responsem->error))
        {
        }
        else
        {
```

```
for($j=0;$j<count($responsem->documents); $j++)
{
        $doc    =    $responsem->documents[$j];
        $mendeley_link    =    $doc->mendeley_url;
        $mendeley_title    =    $doc->title;
        $mendeley_year    =    $doc->year;
        $mendeley_publisher = $doc->publication_outlet;
        $mauthors = $doc->authors;
}
```

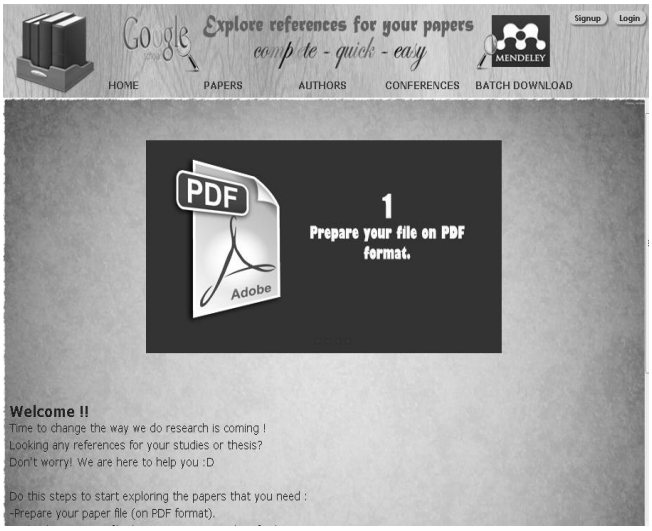
Segmen Program 2. *Source code* untuk mengambil atribut-atribut *paper* yang didapatkan dari hasil pencarian di Mendeley dengan menggunakan Mendeley API.

Fungsi json_decode() adalah fungsi PHP untuk mengubah *file* JSON ke dalam variabel PHP. Jadi variabel $responsem menampung hasil cURL sebagai *response* Mendeley API dari variabel $curl yang telah dideklarasikan dan dieksekusi sebelumnya. Apabila variabel $responsem *error*, maka akan ditampilkan "*No Result*". Sedangkan apabila $responsem menghasilkan *return*, maka dilakukan *looping* untuk setiap *paper* referensi yang ditemukan.

Selanjutnya adalah mengambil nilai atribut-atribut *paper* yang terdapat di dalam $doc. Masing-masing atribut dapat ditampilkan dengan cara $doc->atribut. Jadi apabila dilakukan perintah $doc->title berarti akan ditampilkan judul (*title*) dari *paper* referensi yang berhasil ditemukan melalui Mendeley API.

5. HASIL

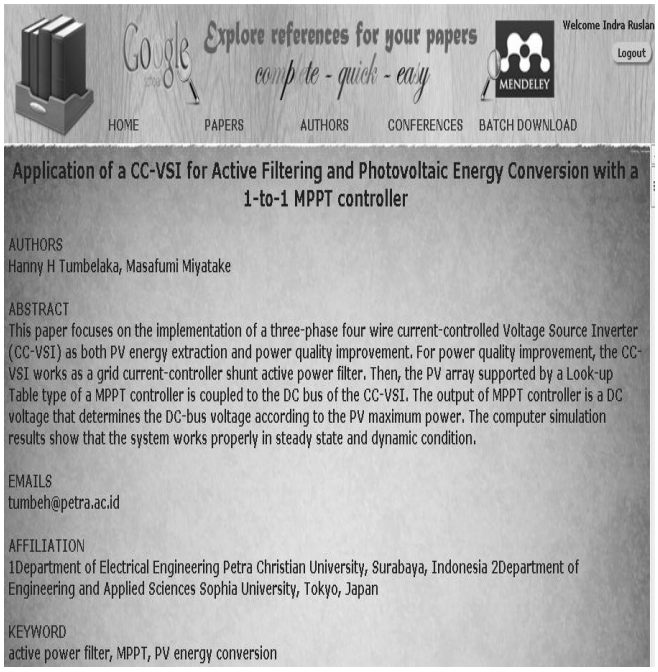
Hasil berupa *website* yang telah di *hosting* pada *server* puslit petra yakni http://puslit2.petra.ac.id dan ber-*domain* pada halaman http://puslit2.petra.ac.id/internal/indra. Gambaran halaman *home* pada *website* dapat dilihat pada Gambar 4.



Gambar 4. Halaman *Home* pada *Website* Penelusuran Artikel Ilmiah.

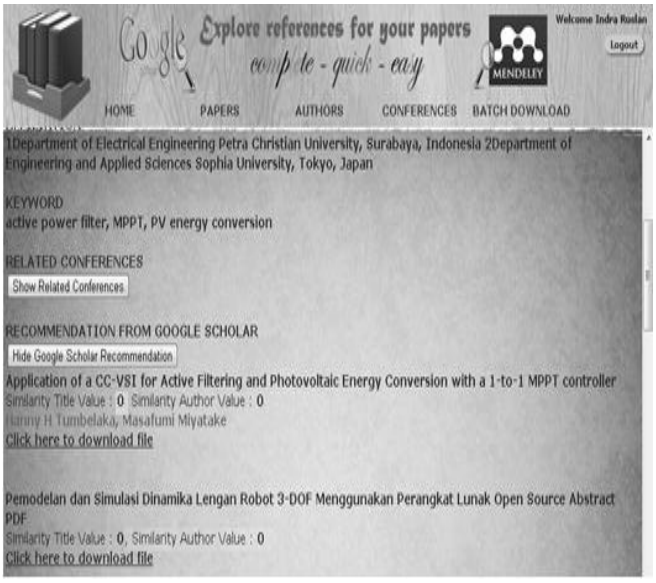
Halaman *Home* merupakan halaman utama yang pertama kali dilihat oleh *user*. Pada halaman ini, *user* akan diberikan penjelasan mengenai tata cara untuk melakukan *upload* terhadap paper ilmiah yang dimilikinya melalui animasi *flash* dan kata-kata. Pada bagian kanan atas halaman *home* terdapat tombol *sign up* dan *login*. *Sign up* merupakan fasilitas bagi *user* yang belum menjadi *member* untuk mendaftar menjadi *member*. Pada halaman *home* terdapat lima buah menu yang dapat dipilih oleh *user*, yakni *home*, *papers*, *authors*, *conferences* dan *batch download*.

Pada halaman *View Paper*, maka akan ditampilkan hasil ekstraksi sitasi oleh ParsCit dari *paper* yang di-*input*-kan oleh *member* seperti yang terlihat pada Gambar 5.

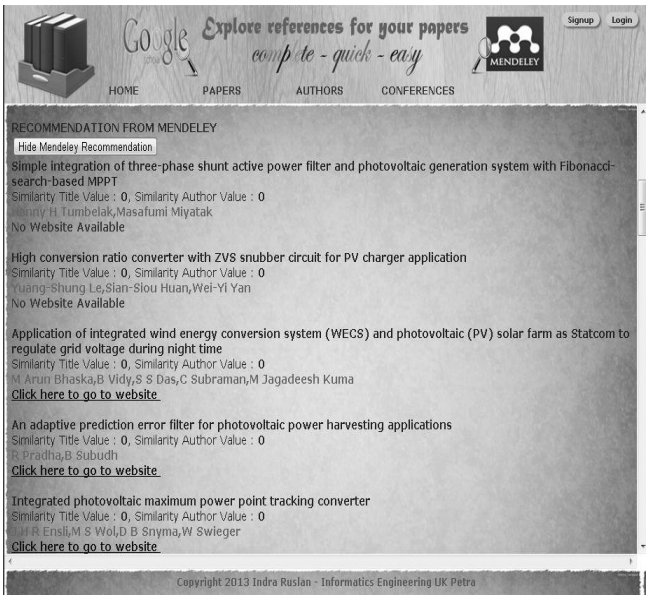


Gambar 5. Halaman *View Paper* pada *Website* Penelusuran Artikel Ilmiah.

Selain itu, juga akan ditampilkan hasil pencarian *paper* referensi di Google Scholar dan Mendeley seperti yang terlihat pada Gambar 6 dan Gambar 7.



Gambar 6. Hasil Pencarian *Paper* Referensi di Google Scholar.



Gambar 7. Hasil Pencarian *Paper* Referensi di Mendeley.

6. KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut:

- Metode *parsing* sudah berhasil mengambil metadata *paper* referensi yang dibutuhkan dari hasil pencarian di Google Scholar dan Mendeley.
- Urutan *paper* yang ditemukan dari hasil pencarian di Google Scholar lebih teratur dibandingkan urutan *paper* yang ditemukan dari hasil pencarian di Mendeley dengan menggunakan Mendeley API.
- Diperlukan perhitungan *similarity* untuk menghitung relevansi dan kemiripan antar *paper* dengan menggunakan proses *ranking* (*scoring*) terutama untuk urutan *paper* yang didapatkan dari hasil pencarian di Mendeley.

7. DAFTAR PUSTAKA

[1] Eyler, A.A., Dreisinger, M. *Publishing on Policy: Trends in Public Health*. Retrieved February 23, 2012 from http://www.cdc.gov/pcd/ issues/2011/jan/09_0247.htm.

[2] Councill, I. G., Giles, C. L., Kan, M.Y. (2008). *ParsCit: An open-source CRF reference string parsing package (Vol.2008, pp.661-667)*. Pennsylvania : European Language Resources Association (ELRA).

[3] Lim, R., Wibowo, A., Sutjiadi, R., Oktian, Y.E. (2012). Pengembangan Paper Citation Extraction Bahasa Indonesia Berbasis ParsCit. In Seminar Nasional Teknologi Informasi (SNTI 2012).

[4] Willinsky, J. (2005). *Open Journal Systems: An example of open source software for journal management and publishing, Journal of Library Hi Tech*, 2005 Vol. 23 (4) ,pp. 504 – 519.

[5] Hegaret, P.L. (2004). *What is the Document Object Model?* Retrieved March 10, 2012, from http://www .w3.org/TR/DOM-Level-3-Core/introduction.html.