

## RANCANG BANGUN SISTEM Pencarian Buku Menggunakan Pengindeks APACHE SOLR

Viny Leilafenci

D3 Manajemen Informatika, Fakultas Teknik, Universitas Negeri Surabaya, vinyleilafenci@mhs.unesa.co.id

I Kadek Dwi Nuryana

Jurusan Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya, dwinuryana@unesa.ac.id

### Abstrak

Ilmu merupakan kebutuhan yang penting bagi manusia untuk meningkatkan kemampuan serta taraf hidupnya. Selain melalui pendidikan formal, ilmu juga dapat diperoleh melalui media cetak atau buku. Buku dapat ditemukan dimana saja, di perpustakaan, di toko buku dll. Meskipun sangat bermanfaat, terdapat kesulitan dalam penggunaan layanan tersebut dikarenakan terlalu banyaknya koleksi pustaka yang ada (buku, jurnal, majalah, dan sebagainya) sehingga sulit untuk menemukan buku yang ingin dicari. Oleh sebab itu, penyedia layanan tersebut harus dapat mengikuti perkembangan zaman yang ada sehingga mempermudah penggunaan layanan tersebut. Penelitian ini membahas tentang sistem yang menghubungkan *user* dengan data buku yang ingin dicari yang berbasis *web* dengan menggunakan teknologi *database MySQL*, *platform pencarian Apache Solr*, dan metode *Performance* pada *FCAPS (Fault, Configuration, Accounting, Performance, Security)* dikarenakan penelitian ini fokus pada hasil performa pencarian buku menggunakan pengindeks *Apache Solr*. Tahap pengujian penelitian ini dibuktikan dengan ketepatan dalam mengindeks buku dan kecepatan dalam pencarian sehingga dapat disimpulkan bahwa *user* dapat menggunakan sistem ini sesuai dengan kebutuhan. Selain itu juga dibuktikan bahwa pencarian menggunakan pengindeks *Apache Solr* lebih cepat daripada menggunakan *MySQL*, baik dari kinerja *CPU* maupun dari saluran *Network*.

**Kata Kunci :** *Web, pengindeks, MySQL, Apache Solr, Performance.*

### Abstract

Knowledge is an important requirement for humans to improve their abilities and living standards. Apart from formal education, knowledge can also be obtained through print media or books. Books can be found anywhere, in the library, in bookstores etc. Although very useful, there are difficulties in using these services because there are too many existing library collections (books, journals, magazines, etc.) making it difficult to find the book user want to find. Therefore, in addition to having to develop by providing library collections, these service providers must be able to keep abreast of the times that exist so as to facilitate the use of these services. This study discusses the system that connects users with data books that want to be searched for web-based using *MySQL* database technology, *Apache Solr* search platform, and *Performance* method on *FCAPS (Fault, Configuration, Accounting, Performance, Security)* due to this research focuses on the results of book search performance using *Apache Solr* indexer. The testing phase of this research is proven by the accuracy in indexing books and speed in searching so that it can be concluded that the user can use this system as needed. It also proved that searching using the *Apache Solr* indexer is faster than using *MySQL*, both from *CPU* performance and from *Network* channel.

**Keywords:** *Web, indexer, MySQL, Apache Solr, Performance.*

### PENDAHULUAN

Ilmu adalah kebutuhan yang penting bagi manusia untuk meningkatkan kemampuan dan taraf hidupnya. Selain melalui pendidikan formal, ilmu juga bisa didapatkan melalui media cetak atau buku. Perpustakaan dan ruang baca merupakan beberapa sarana yang penting dalam menunjang media tersebut. Selain bermanfaat, ternyata masih ada kesulitan dalam penggunaan layanan tersebut karena terlalu banyaknya koleksi pustaka yang ada seperti buku, jurnal, majalah, dan sebagainya sehingga sulit untuk menemukan buku yang ingin dicari. Oleh sebab itu, selain harus berkembang dengan penyediaan buku, harus

dapat mengikuti perkembangan zaman yang ada sehingga mempermudah penggunaan layanan tersebut. Saat ini dalam setiap ruang baca / perpustakaan memiliki koleksi buku yang sangat banyak, baik koleksi fisik maupun digital, dimana daftar koleksi fisik dan dokumen digital dapat diakses melalui *website*. Adanya koleksi pustaka yang sangat banyak ini menyebabkan kesulitan pengguna dalam melakukan proses pencarian.

Berdasarkan permasalahan diatas, maka pada penelitian ini dibuatlah sebuah sistem pencarian buku menggunakan pengindeks *Apache Solr* dan *database MySQL* berbasis *web* yang akan memudahkan *user* dalam mencari buku dengan mudah. Sistem ini juga bisa yang

nantinya disinkronisasikan dengan beberapa ruang baca, perpustakaan, dan toko buku. Selain membahas sistem pencarian, penelitian ini juga menjelaskan kinerja *Apache Solr* dalam ketepatan dan juga kecepatan untuk mengefisiensi waktu pencarian data yang tidak dijelaskan dalam penelitian terdahulu.

*Apache Solr* mempunyai banyak fitur yang bisa digunakan, tetapi pada penelitian ini hanya digunakan sebagai *platform* pencarian. Alasan digunakannya *Apache Solr* sebagai *platform* pencarian yaitu selain sebagai pengindeks, juga dikarenakan waktu *loading Solr* yang lebih efisien dibandingkan dengan pengindeks biasa yaitu *MySQL*. Waktu *loading* yang ditempuh oleh *Solr* lebih singkat walaupun digunakan untuk mencari buku dalam jumlah banyak.

## KAJIAN PUSTAKA

### Penelitian Terdahulu

Studi dalam bidang pencarian menggunakan *Apache Solr* sudah ada pada literatur terdahulu. Literatur – literatur tersebut membuat sebuah aplikasi pencarian menggunakan *Apache Solr*. Penulis pada jurnal (Handojo, Wibowo, Irfanny, Yustivani, & Valentine, 2015) menerapkan sebuah aplikasi *search engine* pada perpustakaan. Aplikasi tersebut bermanfaat untuk mencari buku yang ada pada perpustakaan berbasis *android* menggunakan *platform Apache Solr* dan *database PostgreSQL*.

Selain berbasis *Android*, pada jurnal (Abdenego, Dewi, & Wibowo, 2015) penggunaan sistem pencarian ini bisa diterapkan pada aplikasi berbasis *web*. Aplikasi tersebut digunakan untuk mengolah data seperti data peminjaman / pengembalian buku, data denda, dan data siswa yang berkunjung ke perpustakaan sekolah.

Pada penelitian kali ini tidak hanya membahas sistem pencarian dengan pengindeks *Apache Solr* berbasis *web*, tetapi juga akan menjelaskan bagaimana cara kerja *Solr* mulai dari instalasi, konfigurasi, *query* yang digunakan untuk melakukan pengindeksan, hingga perbandingan performa pencariannya dengan menggunakan *MySQL*.

### Web

Secara umum, *web* adalah suatu layanan yang didapatkan oleh pengguna ketika terhubung ke *internet*. *Web* adalah sebuah perangkat lunak yang berfungsi untuk menerima permintaan atau *request* melalui *HTTP* atau *HTTPS* dari *client* menggunakan *browser* dan dikirimkan kembali atau respons dalam bentuk dokumen *HTML* (Sholichin, 2005). Pengertian lain tentang *web* adalah sebuah sistem informasi melalui *internet*. Informasi yang dikirimkan dapat berupa teks, suara, animasi, gambar, dan *video* yang dapat diakses melalui suatu *software* yang bernama *browser* seperti *Opera*, *Mozilla Firefox*, dan lain sebagainya. Hubungan antara satu halaman *web* dengan halaman *web* yang lainnya disebut *hyperlink*. Sedangkan

*text* yang dijadikan penghubung disebut *hypertext* (Yuhefizar, 2009).

### MySQL

*MySQL* adalah *RDBMS* (atau *database server*) yang mengelola *database* dengan cepat dan menampung data dalam jumlah sangat besar yang dapat di akses oleh banyak *user* (Raharjo, 2011). *MySQL* adalah *database server* yang paling sering digunakan dalam pemrograman *PHP*, *MySQL* digunakan untuk menyimpan data dalam *database* dan memanipulasi data-data yang diperlukan. Manipulasi data tersebut berupa menambah, mengubah, dan menghapus data yang berada dalam sebuah *database* (Nugroho, 2013). Berdasarkan dua pengertian diatas, maka dapat diambil kesimpulan bahwa *MSQL* adalah perangkat lunak basis data yang digunakan untuk mengolah dan menyimpan data dalam skala besar

### Apache Solr

*Solr* (diucapkan "solar") adalah *platform* pencarian perusahaan *open source* yang populer, super cepat, yang ditulis dengan bahasa *Java* dan dibangun dari proyek **Apache Lucene™** pada tahun 2010. *Solr* sangat dapat diandalkan, dapat diukur, dan toleran terhadap kesalahan, menyediakan pengindeksan terdistribusi, replikasi, dan permintaan beban-seimbang, kegagalan dan pemulihan otomatis, konfigurasi terpusat, dan banyak lagi. *Solr* mendukung fitur pencarian dan navigasi dari banyak situs *internet* terbesar di dunia (The Apache Software Foundation, 2017). *Solr* adalah *server* pencarian perusahaan yang berdiri sendiri dengan sebuah *REST* seperti *API*. Dapat meletakkan dokumen di dalamnya (disebut "pengindeksan") melalui *JSON*, *XML*, *CSV* atau biner melalui *HTTP*. Dapat menyanykannya melalui *HTTP GET* dan menerima *JSON*, *XML*, *CSV* atau hasil biner.

### METODE REKAYASA

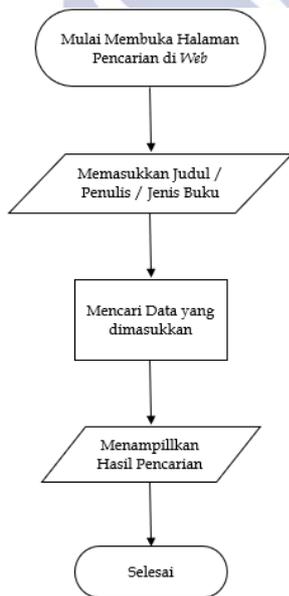
Metode rekayasa yang diterapkan pada penelitian ini yaitu menggunakan metode *FCAPS* (*Fault, Configuration, Accounting, Performance, Security*). Metode *FCAPS* adalah metode yang berfokus pada sebuah penelitian atau pengembangan manajemen jaringan yang digunakan untuk membedakan fungsi-fungsi dari sebuah penelitian atau pengembangan manajemen jaringan (Solusitek, 2014). *Performance* (performa) pada metode *FCAPS* sendiri adalah sebuah fungsi manajemen jaringan yang berfokus untuk memastikan apakah kinerja jaringan yang dibangun tetap pada tingkat yang diharapkan. Metode ini berfungsi untuk memantau kinerja jaringan yang dibangun dan mengidentifikasi sebelum terjadinya masalah dan juga dapat diatur sebagai fungsi *alarm*. Aspek utamanya yaitu pada *response times*, performa (*throughput*), *packet loss rates*, *percentage utilization*, *error rates*, *link utilization*, dan sebagainya.

Berdasarkan penjelasan mengenai metode *FCAPS* diatas, penerapan metode yang digunakan pada penelitian ini yaitu hanya bagian *Performance* karena penelitian ini fokus pada sebuah rancang bangun sistem pencarian buku menggunakan pengindeks *Apache Solr* yang fokus pada performa (*throughput*).

**Alur Sistem**

Sistem ini dirancang untuk menggambarkan alur jalannya *Apache Solr*. Data buku dari contoh *dataset* yang sudah ada yang bertipe *CSV*, akan di *import* ke *MySQL*. Kemudian melakukan *full import* data yang sudah masuk ke *MySQL* ke *Apache Solr*. Di *Apache Solr*, di setting informasi apa yang akan ditampilkan kepada *user* di halaman *web*. Setelah itu membuat sebuah *web* sederhana yang akan digunakan untuk menampilkan data buku yang dicari dalam bentuk *JSON* baik lewat judul, penulis, ataupun jenis buku. Dalam *web* tersebut nantinya akan dilakukan dua (2) macam pencarian, yaitu pencarian yang menggunakan *MySQL* dan pencarian yang menggunakan *Apache Solr*. Setelah itu membandingkan performa kecepatan kedua pencarian tersebut.

Berikut *flowchart* dari pencarian buku di dalam *web* tersebut.



Gambar 1. *Flowchart* Pencarian Buku

Penjelasan *flowchart* diatas yaitu, langkah pertama *user* membuka halaman *web*, lalu masuk ke halaman pencarian. Masukkan judul / penulis / jenis buku yang ingin dicari pada kotak pencarian. Sistem akan mencari buku berdasarkan data yang dimasukkan. Kemudian sistem menampilkan hasil pencarian kepada *user*.

**Kebutuhan Perangkat**

a) Perangkat Keras

Tabel 1. Spesifikasi *Laptop*

<b>Sistem Operasi</b>	Windows 10 Pro 64-bit
-----------------------	-----------------------

<b>Processor</b>	Intel(R) Core(TM) i7-4720HQ CPU @ 2.60 GHz (4 CPUs), ~2.59 GHz
<b>RAM</b>	4 GB
<b>HDD</b>	1 TB
<b>VGA</b>	NVIDIA GeForce 840M

1) Sebuah *laptop* yang digunakan untuk mencari data buku dan membandingkan kecepatan pencarian dari *MySQL* dan *Apache Solr*. Gambar di atas adalah tabel spesifikasi *laptop* yang digunakan untuk membuat penelitian ini.

b) Perangkat Lunak

- 1) Menggunakan *XAMPP* sebagai *server* untuk lokal dalam pembuatan *web* sederhana dan *database* dengan *MySQL*.
- 2) Menggunakan *PhpStorm* untuk membuat halaman *web* sederhana.
- 3) Menggunakan *Java* sebagai salah satu bahasa *script* yang digunakan untuk membuat halaman *web*.
- 4) Menggunakan *Google Chrome* untuk membuka halaman *web* dan mengecek kecepatan *loading* halaman saat melakukan pencarian.
- 5) Menggunakan *extension* dari *Google Chrome* yaitu *Moesif Origin & CORS Changer* untuk memperbolehkan *web* mengakses data yang ada di *Apache Solr*.
- 6) Menggunakan *MySQL* dan *MySQL JDBC* sebagai *software* sistem manajemen basis data.
- 7) Menggunakan *Apache Solr* sebagai *platform* pencarian data buku.

Selain itu diperlukan juga sebuah tabel *database* yang berisi data buku. Tabel ini berisi beberapa deskripsi dari sebuah buku. Berikut adalah struktur tabel data buku:

Tabel 2. Struktur Tabel Buku

No.	Nama Field	Tipe Data	Keterangan
1.	id	integer(15)	Berisi kode buku
2.	last_updated	timestamp	Berisi tanggal update buku
3.	file_name	varchar(100)	Berisi nama file buku
4.	url_pic	varchar(500)	Berisi nama URL gambar buku
5.	book_tit	varchar(500)	Berisi judul buku
6.	writer	varchar(500)	Berisi nama penulis buku
7.	book_num	varchar(50)	Berisi jumlah buku

8.	book_type	varchar(100)	Berisi tipe buku
----	-----------	--------------	------------------

Tabel tersebut merupakan entitas - entitas dari sebuah *database* yang bernama *ta* yang memiliki delapan (8) struktur yang telah terisi data buku yang nantinya hanya beberapa *field* yang akan diindeks di halaman *web*.

**Implementasi Sistem**

Pada penelitian ini menerapkan sistem pencarian yang dimulai dari *user*, pencarian dengan sistem, hingga dikembalikan lagi ke *user* dengan alur sebagai berikut ;



**Gambar 2.** Alur Implementasi Sistem

Berikut ini penjelasan dari alur implementasi sistem tersebut:

- 1) Menyiapkan kebutuhan perangkat lunak yang diperlukan dalam membangun sistem tersebut seperti *XAMPP, PhpStorm, Java, dan Apache Solr*.
- 2) Melakukan instalasi *Java, Apache Solr, dan MySQL JDBC*.
- 3) Melakukan *import* data dari *dataset* yang telah disiapkan, baik kedalam *MySQL* maupun kedalam *Apache Solr*.
- 4) Membuat sebuah *web* sederhana yang akan menampilkan data buku.
- 5) Melakukan pengujian pencarian data buku dengan menggunakan *Apache Solr*.
- 6) Membandingkan kecepatan *loading* pencarian data antara *MySQL* dengan *Apache Solr*.

**Skenario Uji Coba**

Skenario pengujian dari sistem yang telah dibangun yaitu setelah melakukan instalasi perangkat lunak terlebih dahulu. Setelah itu melakukan instalasi dan konfigurasi *MySQL* dan *Apache Solr*. Selanjutnya akan dilakukan pengujian dimana terdapat 2 pengujian yaitu pengindeksan

buku menggunakan *MySQL* dan *Apache Solr* serta perbandingan performa *loading* pencarian data menggunakan *Apache Solr* dan dengan menggunakan *MySQL*. Pada skenario pengujian kali ini memanfaatkan fitur *extension Google Chrome, Moesif Origin & CORS Changer*, yang akan dilakukan sebagai tahap analisa dari uji coba sistem.

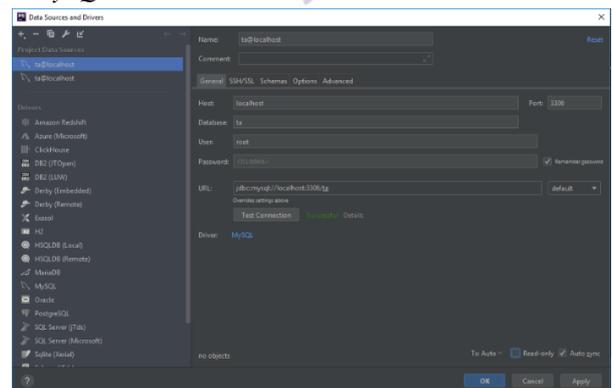
**HASIL DAN PEMBAHASAN**

Sebelum memulai pembahasan, lakukan instalasi perangkat lunak yang dibutuhkan yang sudah disebutkan pada *sub-bab* kebutuhan perangkat.

**Konfigurasi MySQL Server sebagai Manajemen Basis Data**

Setelah melakukan instalasi perangkat lunak yang dibutuhkan, Kemudian yaitu melakukan instalasi *MySQL* yang nantinya digunakan sebagai pembanding performa pencarian dengan *Apache Solr*. Kemudian jalankan *XAMPP* yang sudah di instal, lalu jalankan *Module Apache* dan *MySQL*. Selanjutnya melakukan konfigurasi *MySQL*. Berikut tahap – tahap konfigurasinya:

- 1) Buat *web* sederhana.  
*Web* ini akan digunakan untuk menampilkan kinerja sistem yang terdiri dari halaman awal, halaman daftar buku, halaman pencarian dan halaman deskripsi buku.
- 2) Buat sebuah *database MySQL*.  
*Database* tersebut berisi sebuah tabel, dimana tabel tersebut memiliki delapan (8) entitas seperti pada tabel 2.
- 3) *Import dataset*.  
Lakukan *import dataset* buku yang sudah didownload dari [Github https://raw.githubusercontent.com/uchidalab/book-dataset/master/Task1/book30-listing-train.csv](https://raw.githubusercontent.com/uchidalab/book-dataset/master/Task1/book30-listing-train.csv) yang sudah berbentuk *CSV*.
- 4) Sambungkan *database* dengan *web* yang sudah dibuat. Tujuannya agar *web* bisa menampilkan data yang ada di *database*. Isikan nama *database, user, dan password* (jika ada) lalu klik *Test Connection* untuk mencoba apakah sudah tersambung dengan *database MySQL*.



**Gambar 3.** Konfigurasi *MySQL*

- 5) Buat *file* koneksi *MySQL*.

Pada direktori *web*, tambahkan sebuah *file PHP* untuk memanggil dan membuat sambungan *database MySQL* yang berisikan *servername web* yang akan berjalan, *username* dari kepemilikan *database*, *password*, dan nama dari *database* yang sudah dibuat.

6) Buat *file* daftar buku.

Pada *file* ini berisi kode program untuk menampilkan daftar buku dalam bentuk tabel yang diambil dari *database MySQL* yang sudah dibuat sebelumnya dengan ketentuan mengurutkan data berdasarkan *id* buku yang menampilkan *id*, judul, penulis, dan tipe buku.

**Instalasi dan Konfigurasi Apache Solr sebagai Platform Pencarian Data**

Setelah melakukan konfigurasi *MySQL* selanjutnya melakukan instalasi dan konfigurasi *Apache Solr*. Berikut ini langkah – langkah instalasi *Apache Solr* :

1) *Download* dan *unpack* direktori *Apache Solr*.

*Apache Solr* dapat di *download* di situs resminya. Karena *file download* dalam bentuk *.zip* maka harus di ekstrak terlebih dahulu.

2) Jalankan *Apache Solr*.

Buka *Command Prompt*, masuk ke direktori dimana hasil ekstrak sebelumnya berada. Masuk ke direktori *bin* lalu jalankan *Apache Solr*.

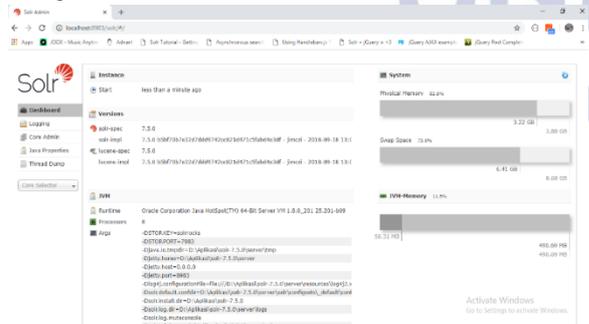
```
D:\Aplikasi\solr-7.5.0\bin> solr
start
```

3) Buat *core* pada *Apache Solr*.

Setelah *Solr* dijalankan, buat sebuah *core* yang digunakan untuk menyimpan data yang diimport dari *MySQL*.

```
D:\Aplikasi\solr-7.5.0\bin> solr
create_core -c [nama core]
```

Selanjutnya buka halaman *Solr Admin*, lalu pilih *Core Selector* dan pilih nama *core* yang telah dibuat untuk mengecek bahwa *core* sudah berhasil dibuat.



Gambar 4. Tampilan *Solr Admin*

Setelah melakukan instalasi, selanjutnya lakukan konfigurasi *Apache Solr*. Berikut langkah – langkahnya :

1) Buat deklarasi koneksi ke *MySQL*.

Buka direktori *solr-7.5.0/server/solr/data\_buku/conf/solrconfig.xml* lalu tambahkan kode program berikut :

```
<lib dir="{solr.install.dir:../../..}/dist/" regex="solr-
dataimporthandler-.*\.jar" />
<lib dir="{solr.install.dir:../../..}/dist/" regex="mysql-
connector-java-.*\.jar" />

<requestHandler name="/dataimport"
class="org.apache.solr.handler.dataimport.DataImportHandler">
  <lst name="defaults">
    <str name="config">data-config.xml</str>
  </lst>
</requestHandler>
```

Gambar 5. Kode Tambahan pada *File solrconfig.xml*  
Kode diatas adalah konfigurasi untuk melakukan koneksi ke *MySQL* menggunakan *DataImportHandler* dengan *file* konektornya bernama *data-config.xml*.

2) Buat *file* konektor ke *MySQL*.

Selanjutnya buat sebuah *file* baru dengan nama *data-config.xml* di direktori *solr-7.5.0/server/solr/data\_buku/conf/*, lalu tambahkan kode program yang berisi *URL MySQL* beserta *username* dan *password*-nya. *Tag entity* berarti satu buah entitas yang diindeks ke dalam *Solr*. Dalam *entity* “*data\_buku*” tersebut terdapat 4 *field* yang akan dimasukkan ke indeks *Solr* yaitu *id*, *book\_tit*, *writer*, dan *book\_type*. Pastikan namanya sama seperti *field* yang ada di *database MySQL* agar lebih mudah.

```
<?xml version="1.0" encoding="UTF-8" ?>
<dataConfig>
  <dataSource type="JdbcDataSource" driver="com.mysql.jdbc.Driver"
  uri="jdbc:mysql://localhost/ta" user="root" password="" />
  <document>
    <entity name="data_buku" query="SELECT id, book_tit, writer,
    book_type FROM data_buku">
      <field column="id" name="id" />
      <field column="book_tit" name="book_tit" />
      <field column="writer" name="writer" />
      <field column="book_type" name="book_type" />
    </entity>
  </document>
</dataConfig>
```

Gambar 6. Kode Koneksi *Solr* ke *MySQL*

3) *Restart Solr*.

Dilakukan dengan cara menghentikan *Solr* dengan mengetik *stop solr -all* pada *Command Prompt* lalu jalankan kembali. Gunanya agar *Solr* dapat membaca perintah yang baru saja dimasukkan.

4) Buat *field* pada *Solr Admin*.

Langkah selanjutnya yaitu buat *field* sebanyak 4 yang masing – masing terdiri dari *id*, *book\_tit*, *writer*, dan *book\_type*. Pilih menu *Schema* kemudian *Add Field*. Nama *field* disesuaikan dengan *field name* pada *file data-config.xml*. *Field type* : *text\_general* agar bisa diindeks dari data yang akan dicari. Centang *stored* dan *indexed*.

5) Lakukan *full import* pada *Solr*.

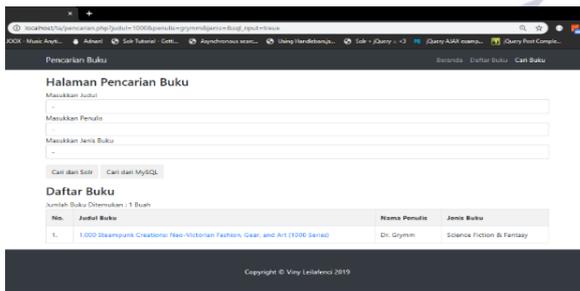
Setelah konfigurasi selesai, Langkah terakhir yaitu melakukan *full import* data dari *MySQL*. Pada *Solr Admin*, pilih *core* yang sudah dibuat, pilih *Dataimport*, pilih *Entity* yang sudah dibuat, lalu klik *Execute*. Jika berhasil, akan muncul pemberitahuan bahwa *indexing completed* dengan jumlah total banyaknya data yang masuk.

**Pengujian Pengindeksan Data Menggunakan *MySQL* dan *Apache Solr***

Selanjutnya yaitu uji coba pencarian buku menggunakan pengindeks *MySQL* dan *Apache Solr*.

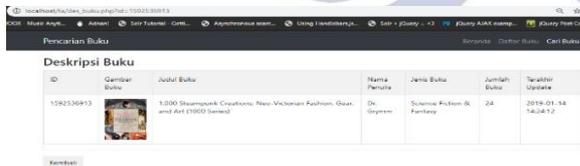
1) Pengindeksan Data Menggunakan *MySQL*

- a. Buat *file* untuk halaman pencarian data buku. Tambahkan kode program untuk mencari data berdasarkan judul / penulis / jenis buku dari *database MySQL*. Kode program tersebut akan memproses pengindeksan data berdasarkan judul / penulis / jenis buku yang dimasukkan oleh *user*. *MySQL* akan menampilkan hasil pencarian dengan ketentuan antara lain diurutkan berdasarkan judul buku serta menghitung jumlah buku yang ada dari kata yang dimasukkan oleh *user*. *User* bisa mencari 2 atau 3 pencarian sekaligus, misalnya judul dan penulis buku, judul dan jenis buku, atau penulis dan jenis buku. Berikut contoh hasil pengindeksan menggunakan *MySQL*:



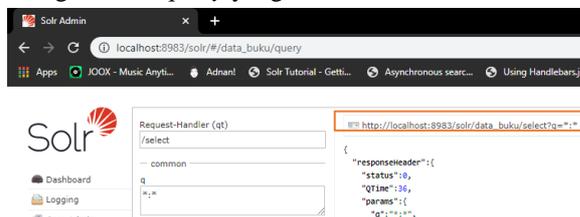
Gambar 7. Hasil Pengindeksan pada *MySQL*  
 Pengindeksan dengan *MySQL* akan memunculkan *query* yang dieksekusi pada *Address Bar*.

- b. Buat *file* untuk halaman deskripsi buku. Setelah melakukan pencarian, *click* judul buku untuk mengetahui deskripsi tentang buku yang dicari karena sebelumnya pada judul buku sudah diberi *link* yang disambungkan ke halaman deskripsi buku. Berikut tampilan halaman deskripsi buku dari judul buku yang dipilih.



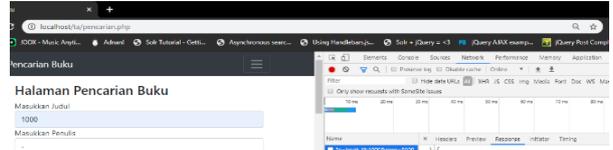
Gambar 8. Tampilan Deskripsi Buku

- 2) Pengindeksan Data Menggunakan *Apache Solr*
  - a. Menambahkan kode pencarian *Solr* pada *file* pencarian. Pada pencarian buku menggunakan pengindeks *Solr*, *query* yang digunakan yaitu *URL* yang tersedia di *Solr Admin* untuk memanggil data yang sudah diimportkan ke dalam *Solr* sebagai ganti dari *query* yang dieksekusi di *address bar*.



Gambar 9. Asal *Query* dari *Solr Admin*

*Query* yang digunakan dapat dilihat dengan melakukan *inspect* halaman dengan cara klik kanan pada halaman pencarian atau dengan menekan *Ctrl+Shift+I* pada *keyboard*, lalu pilih panel *Network*. Kemudian lakukan pencarian dengan memasukkan judul / penulis / tipe buku.



Gambar 10. *Query* yang Digunakan untuk Pengindeksan pada *Solr*

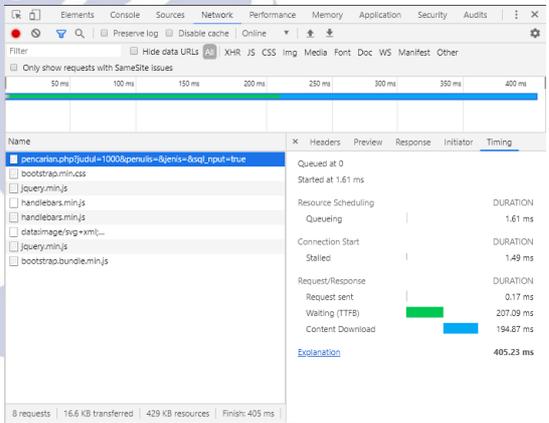
**Hasil Perbandingan Performa Loading Pencarian *MySQL* dan *Apache Solr***

Pada uji coba ini menggunakan 2 panel dari *Google DevTools* yaitu dari *Network* dan dari *Performance*.

- 1) *Laoding* Data Menggunakan *MySQL*

- a. Pada Panel *Network*

Lakukan *inspect* halaman. Pilih panel *Network*, setelah itu baru melakukan pencarian, lalu klik pada *query* pencarian.

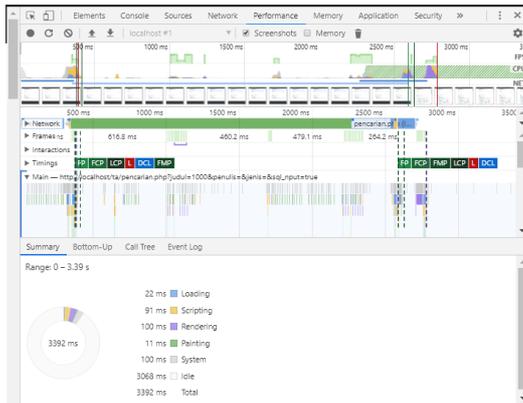


Gambar 11. *Loading* Data di Panel *Network* pada *MySQL*

Berdasarkan gambar 11 dijelaskan bahwa pada saat melakukan uji coba, diperlukan sebanyak 0.17 *ms* untuk *Request Send* atau waktu mengirim permintaan data, 207.09 *ms* (*milliseconds*) *Waiting (TTFB)* / *Time to First Byte* atau waktu untuk *browser* harus menunggu sebelum menerima data, dan sebanyak 194.87 *ms* untuk *Content Download* yaitu waktu *browser* menerima respon. Waktu total yang diperlukan untuk mencari data pada *MySQL* ini sebanyak 405.23 *ms*. Menurut *Google's PageSpeed Insights*, waktu respon *server* seharusnya berada di bawah 200ms (*Developers*, 2018).

- b. Pada Panel *Performance*

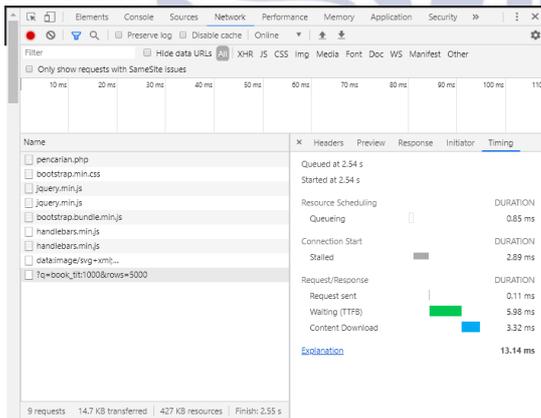
Lakukan *inspect* halaman. Pilih panel *Performance*, setelah itu baru melakukan pencarian, lalu klik pada *query* pencarian.



Gambar 12. Loading Data di Panel Performance pada MySQL

Pada gambar 12 membahas performa loading pencarian data pada MySQL dengan panel Performance. Grafik tersebut merupakan kinerja CPU untuk mencari sebuah data hingga menampilkannya kepada user. Dijelaskan bahwa pada saat melakukan uji coba, diperlukan sebanyak 22 ms untuk loading, 91 ms untuk scripting, 100 ms untuk rendering, 11 ms untuk painting, 100 ms untuk system, dan 3068 ms untuk idle. Total waktu yang dibutuhkan untuk mencari data pada MySQL ini sebanyak 3392 ms.

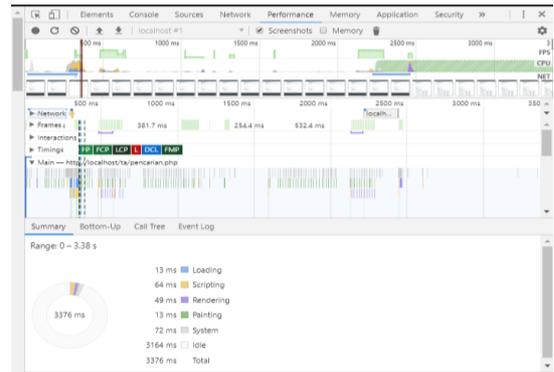
2) Loading Data Menggunakan Apache Solr  
a. Pada Panel Network



Gambar 13. Loading Data di Panel Network pada Solr

Pada gambar 13 membahas performa loading pencarian data pada Solr dengan panel Network. Disini dijelaskan bahwa pada saat melakukan uji coba diperlukan sebanyak 0.11 ms untuk Request Send atau waktu mengirim permintaan data, 5.98 ms Waiting (TTFB), dan sebanyak 3.32 ms untuk Content Download. Waktu total yang diperlukan untuk mencari data pada Solr ini sebanyak 13.14 ms. Jika dibandingkan dengan MySQL yang sebanyak 405.23 ms, maka dapat dikatakan Solr lebih cepat dalam loading data menurut panel Network.

b. Pada Panel Performance



Gambar 14. Loading Data di Panel Performance pada Solr

Pada gambar 14 membahas performa loading pencarian data pada Solr dengan panel Performance. Pada grafik tersebut merupakan kinerja CPU untuk mencari sebuah data hingga menampilkannya kepada user. Dijelaskan bahwa pada saat melakukan uji coba, diperlukan sebanyak 13 ms untuk loading, 64 ms untuk scripting, 49 ms untuk rendering, 13 ms untuk painting, 72 ms untuk system, dan 3164 ms untuk idle. Total waktu yang dibutuhkan untuk mencari data pada Solr ini sebanyak 3376 ms. Jika dibandingkan dengan MySQL yang sebanyak 3392 ms, maka dapat dikatakan Solr lebih cepat dalam loading data menurut panel Performance.

PENUTUP

Simpulan

Setelah melakukan uji coba di atas, maka dapat diambil simpulan bahwa implementasi sistem pencarian menggunakan pengindeks MySQL dan Apache Solr mempunyai beberapa perbedaan dan persamaan, berikut beberapa perbedaannya:

Tabel 3. Perbedaan MySQL dan Apache Solr

Perbedaan	MySQL	Apache Solr
Query yang digunakan	Muncul di Address Bar.	Tidak muncul di Address Bar tetapi bisa dilihat ketika melakukan inspect halaman.
Performa Loading pada Panel Network	-	Lebih cepat beberapa millisecond dari MySQL.
Performa Loading pada Panel Performance	-	Lebih cepat beberapa millisecond dari MySQL.

Efek pada laptop	Membuat laptop agak lemot karena banyak data yang di-load	Tidak membuat laptop lemot karena hanya menampilkan apa yang diindekskan.
------------------	---	---

*Sistem Joomla (CMS) Edisi Revisi*. Jakarta: PT. Elex Komputindo.

Sedangkan persamaannya yaitu ketepatan dalam pengindeksan sesuai, antara *MySQL* dan *Solr*.

### Saran

Penelitian selanjutnya dapat mengimplementasikan sistem ini dengan menggabungkan dengan fitur perpustakaan, toko buku, dan sistem yang membutuhkan sistem pencarian. Akan lebih baik lagi jika dikembangkan bukan hanya berbasis *web*, tetapi berbasis *android* dan *ios*.

### Ucapan Terima Kasih

Ucapan terima kasih kepada **Bapak I Kadek Dwi Nuryana, S.T., M.Kom.** selaku Pembimbing untuk penelitian dan pengerjaan artikel ini.

### DAFTAR PUSTAKA

Abdenego, Dewi, L. P., & Wibowo, A. (2015). *Perancangan dan Pembuatan Sistem Informasi Perpustakaan Berbasis Web pada SMP YBPK 1 Surabaya*.

Developers, G. (2018, 12 14). *Improve Server Response Time*. Retrieved from Google Developers: <https://developers.google.com/speed/docs/insights/Server>

Handojo, A., Wibowo, A., Irfanny, M., Yustivani, A., & Valentine, F. (2015). Aplikasi Search Engine Perpustakaan Petra Berbasis Android dengan Apache SOLR. *Jurnal Energi dan Manufaktur*, <https://ojs.unud.ac.id/index.php/jem/article/view/22792/0>.

Nugroho. (2013). *Mengenal XAMPP Awal*. Yogyakarta: MediaKom.

Raharjo, B. (2011). *Membuat Database Menggunakan MySQL*. Bandung: Informatika.

Sholichin, A. (2005). *Pemograman Web dengan PHP dan Mysql*. Jakarta.

Solusitek. (2014). *Lima Fokus Manajemen Jaringan (FCAPS)*. Retrieved Maret 26, 2018, from <http://www.solusitek.com/?p=3723>

The Apache Software Foundation. (2017). *Solr*. Retrieved from <http://lucene.apache.org/solr/>

Yuhefizar, H. M. (2009). *Cara Mudah Mengembangkan Website Interaktif Menggunakan Content Manajemen*