UNIVERSITAS INDONESIA

IMPLEMENTASI SISTEM SITASI JURNAL ELEKTRONIK INDONESIA BERBASIS TEKNIK EKSTRAKSI WEB

SKRIPSI

AGUNG KURNIAWAN
0706198991

FAKULTAS TEKNIK UNIVERSITAS INDONESIA
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI 2009

UNIVERSITAS INDONESIA

IMPLEMENTASI SISTEM SITASI JURNAL ELEKTRONIK INDONESIA BERBASIS TEKNIK EKSTRAKSI WEB

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana

AGUNG KURNIAWAN
0706198991

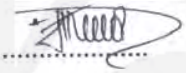
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI 2009

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.

Nama : Agung Kurniawan

NPM : 0706198991

Tanda Tangan : 

Tanggal : 17 Juni 2009

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama	: Agung Kurniawan
NPM	: 0706198991
Program Studi	: Teknik Elektro
Judul Skripsi	: Implementasi Sistem Sitasi Jurnal Elektronik Indonesia Berbasis Teknik Ekstraksi Web

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing	: Dr. Ir. Riri Fitri Sari M.Sc., MM.	(.....)
Penguji	: Prof. Dr. Ir. Bagio Budiardjo MSc.	(.....)
Penguji	: Dr. Ir. Kalamullah Ramli M.Eng.	(.....)

Ditetapkan di : Depok

Tanggal : 24 Juni 2009

UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Allah SWT, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

Dr. Ir. Riri Fitri Sari M.Sc., MM.

Selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini. Akhir kata, saya berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 17 Juni 2009

Penulis

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Agung Kurniawan
NPM : 0706198991
Program Studi : Teknik Elektro
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

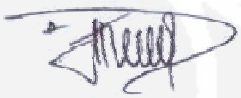
Implementasi Sistem Sitasi Jurnal Elektronik Indonesia Berbasis Teknik Ekstraksi Web

Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam untuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 17 Juni 2009

Yang menyatakan



(Agung Kurniawan)

Universitas Indonesia

ABSTRAK

Nama : Agung Kurniawan
Program Studi : Teknik Elektro
Judul : Implementasi Sistem Sitasi Jurnal Elektronik Indonesia Berbasis Teknik Ekstraksi Web

Banyaknya karya penelitian yang dihasilkan oleh para peneliti sekarang ini tidak terlepas dari karya - karya penelitian yang dihasilkan sebelumnya. Karya penelitian tersebut banyak yang dipublikasikan baik lewat media cetak ataupun elektronik, dan dengan perkembangan media informasi elektronik seperti Internet, menjadikannya sebagai salah satu media publikasi yang banyak digunakan. Karya penelitian yang banyak dipublikasikan tersebut salah satunya berupa jurnal, Untuk mengetahui keterkaitan antara artikel jurnal dengan artikel jurnal sebelumnya, dapat diketahui dengan melihat sitasi antara artikel-artikel tersebut. Dengan demikian dapat diketahui seberapa sering suatu artikel jurnal disitasi oleh artikel jurnal lainnya.

Untuk membantu mengetahui sitasi antara jurnal yang dipublikasikan di Internet, diperlukan suatu sistem yang secara otomatis mendapatkan data yang diperlukan dari situs penyedia jurnal. Dalam Skripsi ini telah dibuat tools yang dapat mengekstraksi halaman web lalu kemudian memilih data-data yang diperlukan. Selain itu diperlukan suatu database yang digunakan untuk menyimpan data hasil ekstraksi tersebut dan mencari keterkaitannya. Data yang telah diproses dapat dilihat menggunakan suatu antarmuka pengguna yang mempunyai fungsi pencarian data sesuai kata kunci yang dimasukkan oleh pengguna. Sehingga akhirnya secara keseluruhan dan bagian sistem membentuk suatu Mashup.

Sistem ini dibangun dengan menggunakan bantuan bahasa PHP dan database MySQL, setelah mempelajari crawler seperti Openkapow robomaker. Dari hasil pengujian terbukti sistem ini dapat bekerja dengan baik mengekstraksi halaman web penyedia jurnal, termasuk halaman pdf tipe tertentu dan menyimpannya dalam database. Hasil pengujian sistem memperlihatkan analisa masalah waktu program dan memori pada komputer dan juga koneksi ke Internet, juga menampilkan keterkaitan sitasi antar artikel jurnal yang ada.

kata kunci : teknik ekstraksi web, indeks sitasi , jurnal

ABSTRACT

Name : Agung Kurniawan
Study Program : Teknik Elektro
Title : Implementation of Indonesia Electronic Journal Citation System Based on Web Extraction Techniques

All research papers produced by the researchers now are based on previous academic publication produced by the other researchers. There are currently exist many research papers published in electronic-media and new-media. Recently, the improvied technology makes Internet becomes the most widely used media. The research papers published in many forms, one of them is a journal. Relation among journals can be traced though their citations. How many times a journal have been cited in other article is also can be calculated.

In order to know the relation among journals which are published on the Internet, we need a system which can automatically can produce a relationship beetwen article from different journal, from different website. Therefore, to extract website pages and then pick required files automatically has been produce in this work. In addition, it also needs a database to save the extracted files and then find the relations. The data which has already processed could be seen in user interface. The interface has searching function by using key word inputted by users. As a result, the whole system forms a Mashup.

We create an automatic extraction for Indonesian electronic journal system, using data from four (4) universiy e-journal. we built the system using PHP language and MySQL database, after carefully study the algorithm in Openkapow Robomaker. The system can succesfully to extract an information from journal provider web pages, whish include special type of PDF pages, then save them in database. The system generated and finally show the connection and relation among all journals. The testing conducted produce the result of the show that processing time evaluation and memory usage for random number of files. The evaluation result show the execution time is dependent on the number of journal series, volumes, and number of articles on related e-journal sites.

keyword : web extraction, citation index, journal

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan ilmu pengetahuan dan teknologi sekarang ini tidak terlepas dari banyaknya penelitian-penelitian yang banyak dilakukan oleh individu-individu atau institusi-institusi baik dari kalangan pendidikan, swasta, dan pemerintahan. Banyak dari hasil penelitian tersebut berupa tugas akhir, makalah, artikel, berita ataupun jurnal yang dipublikasikan baik di media cetak ataupun media elektronik. Di media elektronik selain melalui radio ataupun televisi bisa juga dipublikasikan melalui internet yang perkembangannya sekarang ini hampir ada di semua tempat terutama di kota-kota besar.

Dengan banyaknya publikasi hasil penelitian tersebut terutama di Internet, maka tidak dapat dipungkiri lagi banyak sekali hasil penelitian dalam bentuk dokumen baik itu makalah, paper, jurnal, artikel yang tersebar di Internet. Sehingga dengan kemudahan mengakses dokumen dan mendapatkannya, banyak dokumen-dokumen yang beredar di internet tersebut kemudian dijadikan bahan acuan atau referensi untuk penelitian-penelitian selanjutnya oleh pihak lain. Sehingga dengan demikian dimungkinkan ada banyak sekali keterkaitan antara dokumen satu dengan lainnya yang tersebar di internet tersebut terutama dalam hal referensi, dimana sudah tentu dokumen yang baru terbit akan mengacu ke dokumen yang telah terbit terlebih dahulu.

Untuk mengetahui keterkaitan suatu dokumen yang dijadikan referensi oleh dokumen lainnya, maka diperlukan suatu cara untuk melihat referensi-referensi yang digunakan oleh suatu dokumen, sehingga diketahui dokumen mana saja yang dijadikan bahan referensi dokumen tersebut. Dengan demikian dapat diketahui jumlah sitasi dan oleh dokumen mana saja suatu dokumen disitasi.

Dengan berdasarkan dokumen yang tersebar di internet, maka untuk dapat mengetahui isi dari dokumen tersebut, khususnya bagian referensi dan mencari keterkaitannya, maka dibutuhkan suatu cara untuk dapat membaca isi dari dokumen yang berada pada halaman suatu web. Cara tersebut umumnya disebut dengan ekstraksi web yaitu suatu teknik untuk mengekstraksi informasi dari

website menggunakan suatu perangkat lunak dengan program tertentu, sehingga isi dari bagian yang ada pada halaman web tersebut dapat terbaca dan diproses untuk dicari keterkaitannya dengan isi pada halaman web lainnnya yang berisi dokumen sebelumnya.

Ekstraksi web ini mentransformasikan suatu isi web yang tidak terstruktur menjadi suatu data yang terstruktur sehingga dapat disimpan dalam suatu lembar kerja tertentu untuk lebih memudahkan dalam pemrosesan selanjutnya.

1.2 Perumusan Masalah

Didasari hal-hal yang telah disebutkan di atas, maka salah satu cara untuk lebih memudahkan dalam proses mengekstraksi dan mengolah data yang didapat dari web yang berupa suatu dokumen, perlu dibangun suatu sistem yang secara otomatis mengekstraksi, mengolah data dari dokumen yang didapat untuk dicari keterkaitannya dengan dokumen lainnya, dalam hal ini dokumen lain yang dijadikan bahan referensi.

1.3 Tujuan

Tujuan yang ingin dicapai dari hasil pembuatan dan penulisan skripsi ini adalah membuat suatu sistem yang dapat membantu dalam proses pencarian keterkaitan kepustakaan (referensi) antara satu dokumen hasil karya penelitian dengan dokumen lainnya yang didapat dari internet.

1.4 Batasan masalah

Dikarenakan keterbatasan waktu dan sumber daya yang ada, dan bervariasinya dokumen yang ada di internet. Maka agar skripsi ini dapat diselesaikan sesuai dengan waktu yang diharapkan maka dibuat beberapa batasan dalam permasalahannya, yaitu dokumen-dokumen yang dicari keterkaitannya adalah dokumen berbentuk artikel jurnal, dokumen tersebut umunya dalam format pdf, jurnal-jurnal tersebut tidak termasuk jurnal-jurnal yang tidak bisa diakses bebas, dan jurnal yang digunakan adalah adalah jurnal-jurnal yang dikeluarkan beberapa institusi pendidikan di dalam negeri yaitu Indonesia.

1.5 Metodologi

Untuk merealisasikan sistem di atas, diperlukan beberapa tahapan dalam proses pengerjaannya, diantaranya yaitu melakukan studi literatur dengan mencari literatur yang berhubungan dengan masalah di atas. Sehingga dengan mempelajari literatur yang ada dapat teridentifikasi hal-hal (*tools*) yang diperlukan untuk pembuatan sistem ini.

Setelah terindentifikasi hal-hal (*tools*) yang diperlukan dalam pembuatan sistem, tahap selanjutnya yaitu mempelajari hal-hal (*tools*) tersebut sehingga semua hal yang diperlukan dapat tersedia dan dapat digunakan dengan baik. Tahapan selanjutnya adalah perancangan dari sistem yang dibuat, dimana dalam perancangan ini dititikberatkan pada alur kerja dari sistem yang dibuat dan pembagian fungsi dari hal-hal (*tools*) yang telah dipelajari.

Selanjutnya dilakukan implementasi sistem sesuai dengan perancangan yang telah dilakukan dengan membuat suatu *mashup* yang dapat menampilkan keterkaitan antara artikel jurnal-jurnal yang dikeluarkan oleh beberapa institusi-institusi yang berada di Indonesia dalam hal sitasi. Kemudian melakukan pengujian sistem yang diimplementasikan sehingga dapat diketahui unjuk kerja sistem yang dibuat dan juga perawatan yang diperlukan agar sistem dapat tetap bekerja dengan baik.

1.6 Sistematika Pembahasan

- BAB 1 PENDAHULUAN

  Membahas mengenai latar belakang masalah, masalah, tujuan, batasan masalah, metodologi dan sistematika pembahasan.
- BAB 2 TEORI PENUNJANG

  Membahas teori-teori penunjang yang akan digunakan pada proses perancangan dan implementasi.
- BAB 3 PERANCANGAN

  Membahas mengenai tahapan-tahapan perancangan yang dilakukan dan proses pengerjaan / implementasi sistem yang dibuat.

- BAB 4 IMPLEMENTASI

  Berisi hasil implementasi, pengujian dan analisis dengan cara membandingkannya dengan tujuan pembuatan sistem.

- BAB V PENUTUP

  Berisi mengenai kesimpulan dari semua tahapan yang dilalui dalam proses pembuatan sistem dan saran-saran untuk pengembangan sistem .

BAB 2
TEORI PENUNJANG

1.7 Teknik Ekstraksi Data Web

Screen scraping adalah suatu teknik dimana suatu program dalam komputer mengutip data dari tampilan keluaran program lain, dan program yang melakukannya disebut *screen scrapper*. Yang membedakannya dengan parsing biasa adalah dimana untuk *screen scraping* ini datanya lebih diperuntukan untuk ditampilkan ke pengguna akhir daripada untuk inputan program lain. *Screen scraping* sering mengabaikan data biner (biasanya foto atau data multimedia) dan format elemennya, sehingga cenderung pada data penting berupa teks [1].

Awalnya *screen scraping* digunakan untuk membaca data teks dari tampilan layar komputer. Hal ini dilakukan dengan membaca terminal memori dan dengan menggunakan port tambahan. Alternatif lainnya menjadikan output port dari suatu komputer menjadi input bagi port komputer lainnya.

Umumnya transfer data antara program dilakukan dengan struktur data yang cocok untuk diproses secara otomatis dengan komputer, seperti pada pertukaran format dan protokol yang berstruktur kaku, didokumentasikan dengan baik, dan minimum ambigu. Seringnya transmisi ini tidak dibaca manusia sama sekali. Tetapi untuk output yang berkebalikan dengan hal di atas seperti label yang berlebih atau komentar yang berlebih atau informasi lainnya yang tidak dapat dilakukan dengan proses otomasi. Akan tetapi, meskipun output yang tersedia adalah sebuah tampilan untuk manusia, *screen scraping* menjadi suatu cara untuk mengerjakan transfer data tersebut.

Screen scraping sering digunakan juga untuk antarmuka antara suatu *legacy* sistem yang tidak kompatibel lagi dengan perangkat keras sekarang, atau antarmuka untuk sistem ketiga yang tidak menyediakan API yang tepat.

Web *Scrapping* atau Web *harvesting* atau ekstraksi data web adalah suatu teknik untuk mengutip data atau informasi dari suatu *website* menggunakan *software* dengan program tertentu. Bisasnya program dalam *software* tersebut mensimulasikan eksplorasi manusia terhadap suatu web dengan menggunakan *low-level* HTTP atau menggunakan *full-fledged web* tertentu seperti internet explorer dan mozilla [2].

Web scraping berhubungan dengan pengindeks-an web yang merupakan suatu teknik universal yang dipakai hampir semua search engine. Perbedaannya web scraping lebih berfokus pada tranformasi dari suatu isi web yang tidak terstruktur, umumnya dalam format HTML menjadi suatu format data terstruktur yang dapat disimpan dan dianalisa pada database atau lembar kerja.

Web scraping juga terkait dengan otomasi web, yang mensimulasikan aktivitas *web browsing* dari manusia menggunakan perangkat lunak komputer. Contoh penggunaannya antara lain[3] :

- Perbandingan harga *online* / katalog produk
- monitoring data cuaca
- deteksi perubahan *website*
- penelitian web
- integrasi data web
- *web content* Mashup
- Mengumpulkan informasi seputar perumahan (nama, lokasi, harga, kontak, dan lain-lain)
- Mengkliping artikel (Judul, Isi, Keywords,sumber referensi, dan lain-lain)
- Otomatisasi situs lelang
- Mengekstraksi undian judi, dan lainnya .

Web *scraping* merupakan area yang cukup banyak dikembangkan. Proses otomasi pengumpulan data atau informasi web merupakan tujuan bersama dari Semantik Web. Pengolahan teks, pengertian semantik, kecerdasan buatan dan interaksi manusia dengan komputer, adalah hal yang banyak diperhatikan.

Web scraping menjadi semacam solusi praktis yang berdasarkan teknologi yang ada meskipun beberapa solusi masih khusus. Karena itu ada beberapa level dari otomasi yang tersedia pada *web scraping* antara lain [2] :

- *Human copy-and-paste*: Sering terjadi bahwa teknologi Web *Scraping* tidak bisa menggantikan manusia dari pemeriksaan manual dan meng-*copy-paste*, Kadang-kadang hal ini dapat menjadi satu-satunya solusi yang ada ketika situs Web secara eksplisit terdapat hambatan untuk mencegah mesin otomatisasi.

- *Text grepping and regular expression matching*: Sebuah pendekatan sederhana namun canggih untuk mengambil informasi dari halaman Web dapat berdasarkan unix grep perintah dan kalimat biasa cocok dengan menggunakan perl atau Python bahasa pemrograman.
- *HTTP programming*: statis dan dinamis halaman Web dapat diambil dengan permintaan HTTP ke *server web* yang jauh (*remote*) menggunakan pemrograman socket.
- *DOM parsing*: dengan menambahkan suatu *full-fledged Web browser*, seperti Internet Explorer atau Mozilla, program dapat mengambil isi dinamis yang dihasilkan dari skrip sisi klien.
- *HTML parsers*: beberapa bahasa query data semi berstruktur, seperti XML query language (XQL) dan hyper-text query language (HTQL), dapat digunakan untuk mem-*parsing* halaman HTML dan untuk mengambil konten dan mentransformasi Web.
- *Web-scraping software*: ada banyak perangkat lunak *web scraping* software yang dapat digunakan untuk solusi web *scraping*. Perangkat lunak tersebut mungkin menyediakan antarmuka untuk merekam web sehingga menghilangkan kebutuhan untuk secara manual menulis kode untuk *web scrapping*, atau beberapa skrip dari fungsi yang dapat digunakan untuk mengekstrak dan mentransformasi isi web, dan antarmuka basis data yang dapat menyimpan data yang diambil ke database lokal.

Untuk mengekstraksi data dari suatu web site perlu dilakukan beberapa hal seperti:

- menemukan halaman HTML sasaran dalam sebuah situs dengan mengikuti *hyperlinks*.
- ekstraksi potongan-potongan data yang relevan dari halamannya,
- penyaringan dan pemrosesan data.

Dalam mendapatkan data yang relevan dari halaman suatu web dapat dilakukan dengan *scanning* bagian-bagian efektif pada sebuah dokumen seperti

kolom kutipan, pengarang, judul buku, tanggal dan lain sebagainya (ini dapat disesuaikan dengan tujuan dan kebutuhan penggunaan teknik ekstraksi web yang dibangun). Ada kalanya informasi-informasi mengenai konten suatu dokumen tidak dapat begitu saja diperoleh. Hal ini bisa disebabkan misalnya informasi penting disajikan dalam dokumen non HTML, seperti yang umumnya digunakan saat ini yaitu PDF dan Flash clips [3].

Bervariasinya dokumen yang ada sehingga dimungkinkan ada dokumen yang tidak mengikuti format yang dapat secara otomatis diketahui maksudnya oleh sistem ekstraksi web. Contohnya dalam penulisan kolom referensi, nama pembuat dokumen, judul, dan tanggal pembuatan yang ditulis berbeda tidak sesuai template standar yang diketahui sistem. Dengan adanya perubahan format atau karena format yang digunakan tidak standar, maka teknik ekstraksi web ini harus dapat mengatasi hal tersebut dengan menyediakan kemungkinan-kemungkinan perubahan format yang dapat terjadi.

1.8 Mashup

Mashup adalah suatu aplikasi web yang menggabungkan data dari satu atau banyak sumber ke dalam satu sistem atau alat yang terintegrasi [20]. Istilah mashup juga bisa berupa file media dijital yang berisi gabungan dari teks, gambar, audio, video, animasi, yang mengkombinasikan kembali atau memodifikasi hasil kerja digital untuk menciptakan hasil kerja tambahan (derivatif). Untuk bidang musik, mashup dapat terdiri dari lagu-lagu yang terdiri dari bagian keseluruhan dari lagu lainnya. Mashup (video) didefinisikan sebagai video yang diedit dari lebih dari satu sumber dan nampak seperti suatu video yang utuh.

Mashup tersebut terdiri dari 2 bagian utama yaitu aplikasi web yang menyediakan layanan baru menggunakan berbagai sumber data yang dimilikinya atua data dari sumber lainnya. Juga sumber data yang dibuat dengan menggunakan API atau protokol lainnya seperti HTTP, RSS, REST, dan lainnya. Mashup diakses oleh *Client* dengan menggunakan web browser untuk menampilkan halaman web yang mengandung Mashup.

Contoh Mashup yang ada adalah adalah penggunaan data kartografis dari Google Maps untuk menambah informasi lokasi ke data real-estate, dengan

membuat layanan web yang baru dan berbeda dari yang sudah ada. Konten Mashup biasanya diambil dari pihak ketiga melalui *Interface* publik atau *Application Programming Interface* (API) atau *Web Services.* Selain itu metode mendapatkan konten dari Mashup meliputi *web feeds* (RSS atau Atom) dan *screen scraping*. Banyak orang bereksperimen dengan Mashup menggunakan Amazon, eBay, Flickr, Google, Microsoft, Yahoo, atau YouTube API, yang menuju kepada terbentuknya editor Mashup. Tabel 2.1 berikut ini merupakan beberapa perbedaan antara Mashup dengan portal.

Tabel 2.1. Perbedaan Portal dengan Mashup [20]

No	Karakteristik	Portal	Mashup
1	Klasifikasi	Teknologi lebih tua, ekstensi ke web server tradisional menggunakan pendekatan yang terdefinisi jelas	Teknik Web 2.0
2	Filosofis/Pendekatan	Pendekatan agregasi dengan membagi peranan web server kedalam 2 fase: markup generation and aggregation dari fragment markup	Menggunakan API yang disediakan situs konten untuk agregasi dan penggunaan kembali konten
3	Ketergatungan Konten	Berorientasi aggregate presentasi markup fragment (HTML, WML, VoiceXML)	Dapat beroperasi dengan XML murni dan juga orientasi presentasi konten (HTML)
4	Ketergantungan lokasi	Lokasi di server	Dapat terjadi pada server atau client
5	Gaya Aggregation	Gaya *Salad bar* (tanpa overlap)	Gaya *Melting point* (Struktur arbitrary)
6	Model Event	Model kejadian Baca dan mengupdate	Operasi CRUD berbasis REST
7	Standar relevan	Perilaku Portlet behaviour diatur dengan standar JSR 168, 286, WSRP	Interchange data PeruXML dengan semantik REST. RSS dan Atom banyak dipakai. Standar spesifik akan terbentuk

Ada beberapa tipe dari Mashup seperti *consumer* Mashup, data Mashup dan juga *business* mashup, dan yang paling umum digunakan adalah *consumer* Mashup yang dipergunakan dan membantu untuk kalangan umum, misalnya pada google map. Mashup yang mengkombinasikan antara beberapa informasi atau media yang sama dari berbagai sumber dan mengintegrasikannya dalam satu sistem disebut juga sebagai data mashup. Selain itu, *business* mashup lebih berfokus pada suatu sistem untuk kerjasama suatu perusahaan.

Secara arsitektur, Mashup ini terbagi menjadi 2 yaitu *Web-based* dan *Server-based*. Mashup berbasis Web umumnya berupa *web browser* pengguna yang melakukan penggabungan dan memformat ulang data. Mashup berbasis Server akan melakukan analisa dan format ulang data yang ada dilakukan pada server dan mengirimkannya ke browser pengguna.

1.9 Sitasi dan Indeks Sitasi

Sitasi merupakan suatu rujukan terhadap suatu buku, artikel, jurnal, halaman web ataupun bentuk-bentuk publikasi lainnya, dengan rincian yang cukup untuk mengidentifikasi sumber rujukan tersebut.

Isi sitasi biasanya terdiri dari nama penulis, judul buku, atau artikel, penerbit, tahun publikasi, dan URL juga tanggal karya tersebut diakses. Selain beberapa hal yang biasanya ada pada sitasi seperti yang disebutkan di atas, ada juga beberapa standar penulisan sitasi yang dibuat dan diterbitkan oleh berbagai asosiasi atau individu yang digunakan oleh penulis, misalnya *Chicago style* dan *Turabian style* yang digunakan untuk semua bidang. *Modern association Language* (MLA) yang digunakan untuk seni, kesusastraan, dan humaniora, *American Psycological Association* (APA) yang digunakan untuk psikologi, pendidikan, dan ilmu sosial lainnya, *American Medical Association* (AMA) untuk bidang kedokteran, kesehatan, dan biologi. Standar penulisan lainnya seperti *National Library Of Medicine* (NLM), *American Chemical Society* (ACS), *American Polotical Science Association* (APSA), *Councils Of Biology Editors* (CBE), *IEEE style, American Sociological Association* (ASA), *columbia style*, dan *Modern Humanities Research Association* (MHRA) [23].

Sitasi atau kutipan terhadap suatu karya ilmiah ataupun dokumen dilakukan karena dokumen yang dikutip tersebut menyediakan informasi yang relevan terhadap penelitian atau tulisan yang dikerjakan oleh penulis. Dengan demikian dapat diketahui bahwa makin sering sebuah dokumen dikutip, maka semakin besarlah dokumen tersebut memberikan kontribusi informasi, dan semakin besarlah pula pengaruhnya pada penelitian atau penulisan yang sedang dilaporkan di dalam dokumen pengutip. Ukuran dari pengaruh atau dampak (*impact*) dari sebuah dokumen memberikan suatu informasi tergantung pada jumlah pengutipan terhadap dokumen tersebut.

Dengan mengetahui berapa kali sebuah dokumen dikutip dalam satu rentang waktu tertentu menunjukkan berapa banyak informasi di dalam dokumen tersebut berguna untuk sebuah penelitian atau penulisan. Dimana apabila frekuensinya menurun, maka dokumen tersebut semakin tidak relevan, sampai akhirnya menjadi usang alias *obsolete*. Apabila terdapat dua dokumen bersama-sama dikutip oleh suatu dokumen, maka kedua dokumen tersebut bersama-sama memberi sumbangan informasi yang saling terkait. Sehingga semakin sering dua dokumen dikutip bersama (*co-cited*), maka semakin dekatlah hubungan kedua dokumen tersebut.

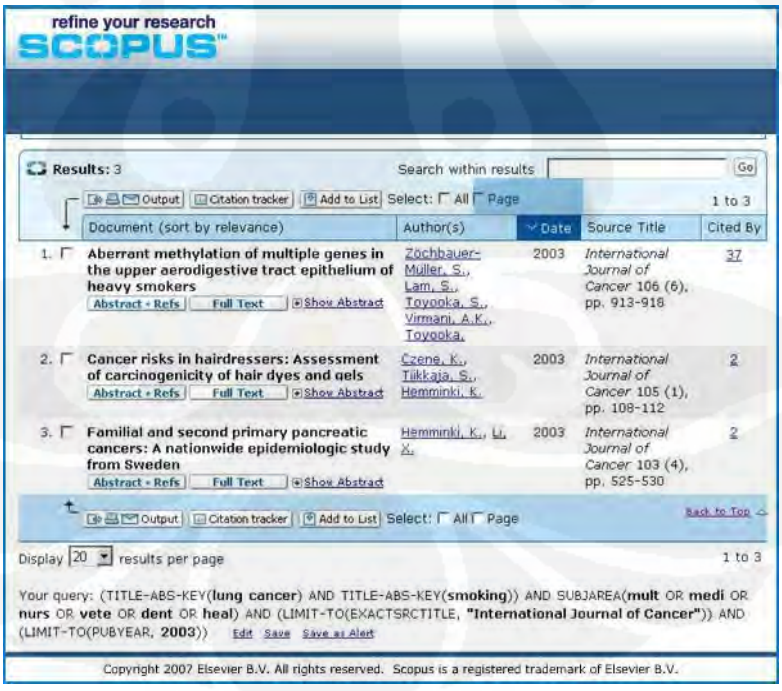
Indeks sitasi merupakan suatu indeks dari sitasi antara publikasi, yang memungkinkan pengguna dapat dengan mudah menentukan dokumen mana yang muncul belakangan dan mensitasi dokumen yang muncul sebelumnya. Sitasi ini telah lama berkembang. Pertama kali pada tahun 1873 terdapat sitasi Shepard yang resmi digunakan. Kemudian tahun 1960, Eugene Garfield dari *Institute For Scientific Information* (ISI) memulai indeks sitasi untuk paper yang diterbitkan di jurnal akademik, diawali dengan *Science Citation Index* (SCI) lalu kemudian *Social Sciences Citation Index* (SSCI) dan *Art And Humanities Index* (AHCI) [22].

Sekarang ini telah banyak yang menyediakan layanan indeks sitasi baik yang berbayar ataupun dapat digunakan secara gratis, bebarapa diantaranya antara lain :

- **ISI** merupakan layanan berbayar yang merupakan bagian dari *Thomson Scientific*. Indeks sitasi ISI diterbitkan dengan dicetak dan dengan CD.

Sekarang ISI bisa diakses lewat web dengan nama *Web of Science*. *Web of science* ini merupakan sebuah layanan akademik *online* yang dapat diakses dengan menggunakan ISI *Web of Knowledge* (WoK), dimana WoK menyediakan akses ke database dan sumber lainnya. Salah satunya adalah *Web Of Science* yang menyediakan indeks sitasi seperti *Science Citation Index* (SCI), *Social Sciences Citation Index* (SSCI), *Arts and Humanities Citation Index* (A&HCI), *Index Chemicus, Current Chemical Reactions, Conference Proceedings Citation Index untuk Science dan Social Science and Humanities.*

- **Scopus** yang diterbitkan oleh Elsevier, merupakan *database* abstrak dan sitasi untuk jurnal, yang menyediakan 15800 jurnal *scientific*, *technical*, *medical* dan *social science* tiap *review*. Scopus juga menawarkan penulis meliputi keanggotaan, jumlah publikasi, data bibliografi, referensi dan jumlah sitasi masing-masing terbitan dokumen. Scopus ini merupakan layanan yang digunakan dengan cara berlangganan / berbayar. Berikut contoh tampilan salah satu halaman web pada Scopus:



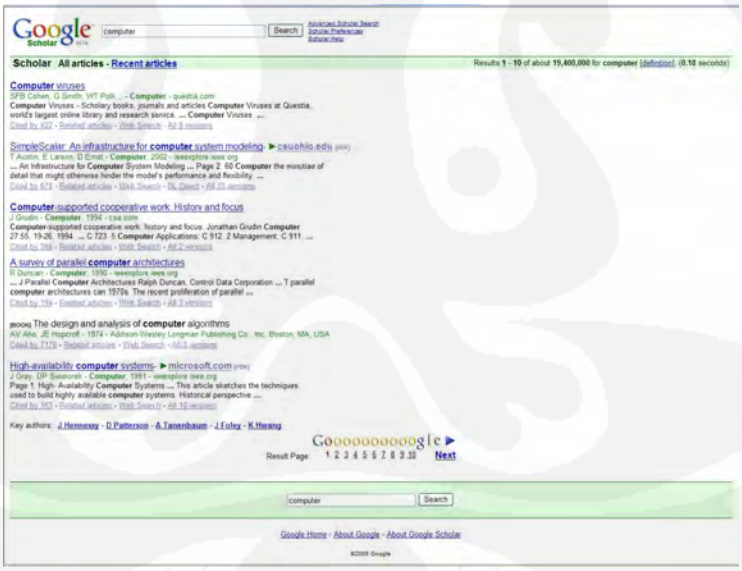
Gambar 2.1. Tampilan Halaman Web Layanan Scopus [25]

- **CiteseerX** merupakan penyedia indeks sitasi yang dapat digunakan gratis untuk *scientific* dan akademik paper yaitu bidang komputer dan ilmu informatika. Dasar dari citeseerX ini adalah citeseer yang kemudian dibuat dengan infrastruktur *open source* baru yaitu Seersuite dan algoritma yang baru pada penerapannya. Sehingga bertujuan melanjutkan citeseer untuk mengambil dokumen *scientific* dan akademik untuk dijadikan indeks sitasi yang kemudian dapat digunakan untuk merangking dokumen dengan menggunakan *impact of citation*. Berikut merupakan tampilan halaman web dari citeseerX :

Gambar 2.2. Tampilan Halaman Web Layanan CiteSeerX [26]

- **RePeC** menyediakan indeks di bidang ekonomi, *Research Papers in Economics* (RePEc) ini merupakan kerjasama dari ratusan sukarelawan dari 57 negara untuk mempertinggi penyebaran penelitian dalam bidang ekonomi. Dengan menggunakan IDEAS *database* yang menyediakan paper, jurnal, informasi penulis dan direktori untuk institusi, RePEC menyediakan lebih dari 700000 artikel lengkap yang dapat diunduh secara gratis.
- **Google Scholar** merupakan salah satu penyedia indeks sitasi yang dapat digunakan gratis. Google Scholar ini dirilis versi beta-nya pada bulan November 2004. Fungsi dari google scholar ini mirip seperti Scopus,

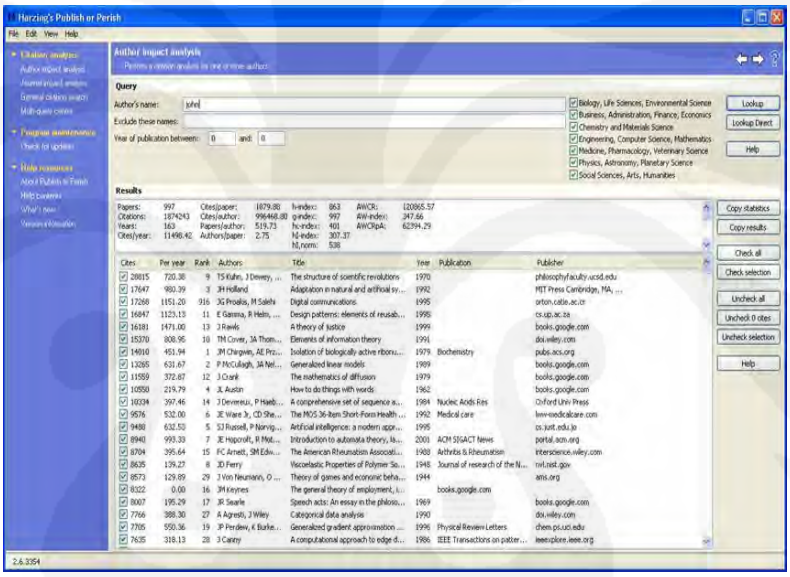
citeseerX, *Web of Science*. Apabila *Thomson Scientific* dan Scopus membuat laporan *Citation Indexes* berdasarkan data primer (dari *database* mereka), maka Google Scholar memanfaatkan artikel-artikel yang tersedia bebas di Internet (umumnya dari artikel serupa yang disimpan dalam website pribadi penulis ataupun *repository* universitasnya) ataupun dari literatur abu-abu seperti buku, *proceeding*, monograf, *website* penulis, dan lain sebagainya. Walau demikian ketepatan perhitungan Google Scholar cukup tinggi, terlebih lagi untuk artikel-artikel yang terbit setelah tahun 2004. Sekarang ini Google Scolar menyediakan hitungan sitasi (*citation count*) yang dapat diakses gratis melalui Internet sehingga semua orang kini dapat menyiapkan laporan *citation count, citation index*, ataupun *impact factor* tanpa harus berlangganan ke jasa-jasa komersial seperti *Thomson Scientific* atau Elsevier. Dengan demikian kemungkinan produk berbayar yang ada sebelumnya bisa saja tidak terpakai lagi dimasa depan. Berikut tampilan halaman web dari Google Scholar:



Gambar 2.3. Tampilan Halaman Web Layanan Google Scholar [27]

- **Publish or Perish** adalah suatu perangkat lunak yang dapat digunakan untuk menganalisa sitasi akademik, dimana data yang digunakan sebagai

dasar perhitungannya menggunakan data sitasi yang ada pada google scholar. Perangkat lunak ini dibuat oleh Harzing.com, yang merupakan *website* professor Anne-Wil Harzing. Professor Harzing bekerja di Departemen *International Management* di University of Melbourne, Australia. Beberapa hasil analisa sitasi yang bisa didapat dari perangkat lunak ini antara lain Hirsch's h-indeks, Egghe's g-indeks, h-indeks sementara, *Age-weighted citation rate* (AWCR), AW-indeks dan beberapa analisa lainnya. Tampilan dari publish or perish dapat dilihat dari gambar berikut :



Gambar 2.4. Tampilan Halaman Publish or Perish

Indeks sitasi ini dapat digunakan untuk merangking karya ilmiah-karya ilmiah yang ada, baik dari tema ataupun penulisnya. Selain itu Publish or Perish dapat dijadikan juga acuan bagi penulis atau peneliti untuk mengetahui informasi terkini mengenai karya ilmiah yang telah dihasilkannya, terutama mengenai berapa jumlah peneliti atau acuan dalam artikel atau hasil karya ilmiah lainnya yang telah mengacu pada hasil karyanya. Sehingga dengan demikian merupakan kebanggaan tersendiri bagi peneliti jika suatu hasil karyanya disitasi oleh hasil karya akademik lainnya, yang berarti telah diakuinya dan diacunya suatu hasil karya tersebut.

Biasanya kesimpulan yang bisa didapat dari adanya indeks sitasi ini yaitu *impact factor* (IF) yang merupakan ukuran dari sitasi (*citation*) terhadap jurnal-jurnal ilmu pengetahuan alam (*science*) dan ilmu pengetahuan social (*social science*) dan sering kali digunakan sebagai ukuran terhadap pentingnya suatu jurnal dalam bidangnya. *Impact Factor* diciptakan oleh Eugene Garfield dari *Institute of Scientific Information* (ISI, kini bagian dari *Thomson Scientific*) pada tahun 1960 dengan menghitung indeks sitasi (*citation index*) dari jurnal-jurnal yang diindeks oleh ISI dan dilaporkan setiap tahun dalam *Journal Citation Report* (JCR) [24].

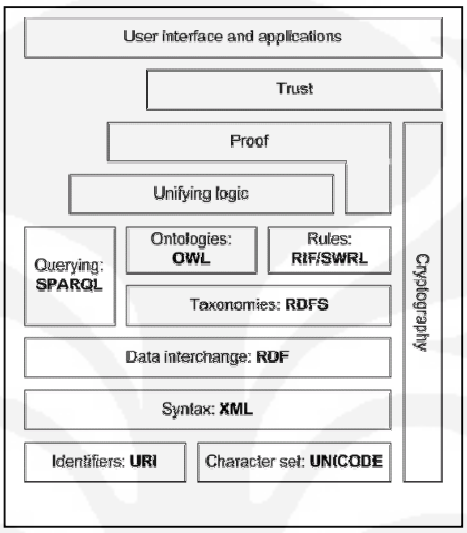
IF ini dihitung dengan menghitung jumlah jurnal yang mengacu ke jurnal yang terbit pada periode dua tahun sebelumnya dan membaginya dengan jumlah jurnal yang diterbitkan pada periode dua tahun sebelumnya tersebut. Akan tetapi banyak debat tentang *impact factor* tersebut, dikarenakan validitasnya, kemungkinan manipulasi dan kesalahan penggunaan. Sehingga ada juga cara lain yang digunakan untuk menilai suatu peneliti atau jurnal seperti menggunakan h-indeks ataupun m-indeks.

1.10 Web Semantik

Web semantik adalah pengembangan *world wide web* yang memungkinkan web untuk memahami dan dapat memenuhi permintaan dari manusia dan mesin untuk menggunakan isi web. Dengan kata lain mengacu kepada kemampuan aplikasi komputer untuk lebih memahami bahasa manusia, sehingga memudahkan pengolahan bahasa dan pengenalan homonim, sinonim, atau atribut yang berbeda pada suatu *database*.

Web semantik itu sendiri diperkenalkan oleh Tim Berners-Lee, penemu *World Wide Web*. Sekarang, prinsip web semantik disebut-sebut akan muncul pada Web 3.0, generasi ketiga dari *World Wide Web*. Bahkan Web 3.0 itu sendiri sering disamakan dengan web semantik. Tujuan dari web semantik adalah web menjadi media yang universal untuk data, informasi dan saling bertukar pengetahuan.

Web semantik terdiri dari standar dan *tools* diantaranya XML, XMLS (XML *Schema*), RDF, *Resources Description Network Schema* (RDFS) dan OWL yang diatur dalam Stak Semantik web seperti pada Gambar 2.5 [4] :



Gambar 2.5. Tumpukan Web Semantik

Masing-masing standar dan *tools* yang ada tersebut mempunyai fungsi dan keterkaitan berikut :

- XML menyediakan elemen sintak untuk isi dari struktur dalam dokumen.
- XML *Schema* adalah bahasa yang menyediakan yang menyediakan dan membatasi isi dan struktur dari elemen dalam dokumen XML.
- RDF adalah bahasa sederhana untuk menyatakan model data yang mengacu ke objek ("*resources*") dan hubungannya, sebuah RDF *based-model* dapat direpresentasikan pada sintak XML.
- RDF *Schema* adalah daftar kosa kata untuk menggambarkan properti-properti dan kelas-kelas dari sumber berbasis RDF.

1. OWL menambahkan lebih banyak daftar kosa kata untuk menggambarkan properti-properti dan kelas-kelas, seperti hubungan antar kelas, keutamaan, persamaan, cara penulisan properti, karakteristik properti dan jumlah kelas.

- SPARQL adalah protokol dan bahasa untuk web semantik data

2.4.2. RDF

Resources Description framework (RDF) adalah standar W3C untuk menggambarkan sumber daya dari web, seperti judul, penulis, tanggal perubahan, isi dan informasi hak cipta dari suatu halaman web. Dengan kata lain RDF ini adalah kerangka untuk membuat *resource* dari web, yang menyediakan suatu model untuk data. RDF ini didesain untuk dimengerti oleh komputer dan tidak didesain untuk ditampilkan ke *user*. RDF ditulis dalam XML sehingga dapat dengan mudah bertukar informasi antara tipe komputer yang berbeda, tipe sistem operasi yang berbeda dan aplikasi bahasa yang berbeda, dan merupakan bagian dari aktivitas web semantik sehingga informasi web mempunyai arti yang tepat, dapat dimngerti dan diproses oleh komputer dan komputer dapat mengintegrasikan informasi dari web.

Contoh penggunaan RDF antara lain untuk membuat properti dari item belanja seperti harga dan stok, jadwal untuk kejadian pada web, informasi halaman web seperti isi dan penulis, rating gambar pada web, mesin pencari, atau pustaka elektronik [7].

RDF menggunakan Uniform Resource Identifier (URI) untuk mengidentifikasi *resource* dimana *resource* dibuat dengan properti dan nilai *property*. *Resource* adalah apapun yang mempunyai URI seperti http://www.w3schools.com/RDF. Properti adalah sebuah *resource* yang mempunyai nama seperti “author” atau “homepage”, dan *property value* adalah nilai dari *property* seperti “Jan Egil Refsnes “ atau http://www.w3schools.com. Seperti pada contoh berikut [6] :

```
<?xml version="1.0"?>
<RDF>
  <Description about="http://www.w3schools.com/RDF">
    <author>Jan Egil Refsnes</author>
    <homepage>http://www.w3schools.com</homepage>
  </Description>
</RDF>
```

- Pernyataan RDF

  Kombinasi dari *Resource*, *Property*, dan *Property value* membentuk suatu pernyataan (menjadi suatu subjek, predikat dan objek dari sebuah pernyataan). Seperti contoh statemen berikut

Untuk pernyataan: "The author of http://www.w3schools.com/RDF is Jan Egil Refsnes".

- Subjek dari pernyataan diatas adalah : http://www.w3schools.com/RDF
- Predikatnya adalah: author
- Objeknya adalah: Jan Egil Refsnes

- Elemen utama RDF

Elemen utama RDF adalah elemen *root*, elemen <RDF>, dan elemen <deskripsi> yang mengidentifikasi *resources.*

Elemen <RDF> adalah root elemen dari dokumen RDF, mendefinisikan dokumen XML dibuat menjadi dokumen RDF, seperti contoh berikut :

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
.
. Description goes here
.
</rdf:RDF>
```

Elemen <deskripsi> mengidentifikasi sebuah *resource* dengan atribut *about.* <rdf:Description> elemen terdiri dari elemen yang menjelaskan *resource* seperti pada contoh berikut :

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cd="http://www.recshop.fake/cd#">
<rdf:Description
 rdf:about="http://www.recshop.fake/cd/Empire Burlesque">
  <cd:artist>Bob Dylan</cd:artist>
  <cd:country>USA</cd:country>
  <cd:company>Columbia</cd:company>
  <cd:price>10.90</cd:price>
  <cd:year>1985</cd:year>
</rdf:Description>
</rdf:RDF>
```

2.4.2. XML

EXtensible Markup Language (XML) adalah sebuah teknologi *cross platform,* dan merupakan tool untuk melakukan transmisi informasi. XML

bukanlah program, atau pustaka. XML adalah sebuah teknologi, sebuah standar dengan berbagai aturan tertentu [10]. Dalam pengertian yang sederhana, sebuah dokumen XML hanyalah sebuah file teks biasa yang berisi berbagai tag yang didefinisikan sendiri oleh pembuat dokumen XML tersebut.

XML sebenarnya bukan teknologi baru, tapi merupakan turunan dari SGML yang telah dikembangkan sejak awal 80-an dan telah banyak digunakan. XML dikembangkan mulai tahun 1996 dan mendapatkan pengakuan dari W3C pada bulan Februari 1998. Sesuai dengan namanya, *eXtensible Markup Language*, sebuah dokumen XML adalah sebuah dokumen dengan *markup*, sama seperti halnya dengan HTML [11] .

Saat ini XML bukan merupakan pengganti HTML. Masing-masing dikembangkan untuk tujuan yang berlainan. Jika HTML digunakan untuk menampilkan informasi dan berfokus pada bagaimana informasi terlihat, sedangkan XML mendeskripsikan susunan informasi dan berfokus pada informasi itu sendiri. Utamanya XML dibutuhkan untuk menyusun dan menyajikan informasi dengan format yang tidak mengandung format standar seperti *heading*, *paragraph*, *table* dan lain sebagainya.

Meskipun tidak terhitung jumlah aplikasi menngunakan XML, berikut beberapa contoh dari *platform* dan aplikasi yang menggunakan teknologi ini [8]:

- Telepon *selular* : data XML dikirimkan ke telepon selular kemudian data tersebut diolah oleh perangkat lunak yang ada pada telepon tersebut untuk kemudian ditampilkan baik menjadi teks, gambar ataupun suara.
- File *converter* : banyak aplikasi yang mengkonversikan dokumen tertentu menjadi format XML.
- *Voice* XML : mengkonversikan dokumen XML menjadi audio format.
-

- Bagian – bagian dokumen XML

Dokumen XML terdiri dari beberapa bagian dan bertujuan agar dokumen tersebut tetap sederhana, berikut beberapa bagian dari dokumen XML :

Prolog
    XML Declaration
    DTD Declaration
Root Element (Document)
    Elements (Nested Elements)

Element Attributes and Values
Data

- XML Prolog

  XML prolog adalah dokumen tambahan dalam XML yang harus ada sebelum *root element*, dimana ada dua bagian yang digunakan untuk prolog yaitu :

  - Deklarasi XML yaitu deklarasi yang menunjukan versi dari XML yang digunakan

    <?xml version="1.0"?>

  - *Document Type Declaration* (DTD) mendefinisikan tipe atau aturan-aturan dari dokumen XML yang dibuat. DTD juga mendefinisikan struktur dokumen XML dengan daftar elemen yang digunakan. DTD memungkinkan format untuk setiap file xml yang unik. Unsur-unsur yang dideklarasikan dalam DTD adalah semua unsur yang membentuk suatu dokumen XML seperti *element, attribut* dan *entity*.

    - Elemen

      Bila sebuah elemen mengandung beberapa elemen anak, maka kita perlu mendeklarasikan elemen anak apa saja yang dipunyai elemen tersebut. Pada contoh berikut kita melihat deklarasi elemen

      ```
      <!ELEMENT organisasi (anggota)>
      <!ELEMENT anggota (nama,alamat)>
      ```

      Dapat dilihat bahwa elemen bernama organisasi memiliki satu elemen anak bernama anggota. Lalu elemen bernama anggota itu sendiri mempunyai dua elemen anak yang bernama nama, dan alamat. Lalu kita perlu medeklarasikan juga type dari elemen-elemen diatas.

      ```
      <!ELEMENT nama (#PCDATA)>
      <!ELEMENT alamat (#PCDATA)>
      ```

      Contoh diatas adalah cara untuk mendeklarasi tipe elemen, dimana element nama, dan semuanya bertipe (`#PCDATA`)

    - Attribute

      Agar dokumen XML valid, perlu didefinisikan semua atribut yang akan digunakan dalam dokumen. Untuk mendefinisikannya digunakan deklarasi daftar atribut. Caranya seperti berikut:

```
<!ATTLIST namaelemen spesifikasiattribut>
```

Nama elemen adalah nama elemen dimana atribut itu digunakan. Sedangkan spesifikasi atribut adalah serangkaian informasi tentang atribut itu. Unsur yang membentuknya antara lain nama atribut, tipe atribut, nilai awal (***default value***), dan sifat atribut. perhatikan contoh dibawah:

```
<!ATTLIST ORGANISASI Nama CDATA #FIXED “HMTE”>
```

maksud dari contoh diatas adalah elemen ORGANISASI memiliki Nama Atribute yang bertipe CDATA. Sifat atribut dalam hal ini adalah #FIXED, yaitu nilai dari atribut nama harus seperti yang dideklarasikan (“HMTE” adalah nilai default yang diberikan bila kita tidak menyebutkannya).

- *Entity*

Entity dapat didefinisikan di dalam DTD sehingga dapat digunakan pada seluruh dokumen XML. Seperti contoh berikut:

```
<?xml version=”1.0” encoding=”iso-8859-1”>
<!DOCTYPE organisasi [
<!ENTITY judul "Manajemen data dan informasi dengan
XML/XSLT"> ]>
```

DTD dapat dibuat menjadi satu dalam satu dokumen XML atau dengan file tersendiri terpisah dari dokumen XMLnya / eksternal. Ada dua tipe deklarasi yang digunakan jika menggunakan eksternal DTD yaitu PUBLIC bila menggunakan aturan yang telah umum, dan SYSTEM jika didefinisikan tersendiri : contoh tipe PUBLIC

```
<!DOCTYPE xml PUBLIC "-//W3C//DTD XML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xml1/DTD/xml1-transitional.dtd">
```

Contoh tipe SYSTEM

```
<!DOCTYPE organisasi SYSTEM "organisasi.dtd">
```

- XML *Root Element (Document)*

Dalam *markup language* elemen pertama yang muncul adalah *root element* yang mendefinisikan jenis dari file dokumen tersebut. *Root element* juga dianggap sebagai dokumen dari XML, atau isi fungsi dari dokumen XML

tersebut. Ketika membuat file HTML maka *tag* <*html*> adalah *root element*, *root element* harus menjadi elemen pertama dari suatu dokumen XML.

Dalam HTML *root element* selalu berbentuk html akan tetapi dalam XML *root element* bisa apa saja. *Root element* ini bisa mempunyai beberapa elemen yang disimpan setelah *tag root* pembuka dan sebelum *tag root* penutup, seperti pada contoh berikut dimana phonebook merupakan root elemen.

```
<phonebook>
   <number>  ... </number>
   <name>  ... </name>
</phonebook>
```

- XML Elemen

XML dokumen terdiri dari elemen-elemen yang terdapat dalam *root* elemen, atau istilah lain element bisa juga disebut sebagai *tag*, dimana masing masing element tersebut merepresentasikan tipe data yang berbeda yang ada di dokumen, misal elemen <*p*> merepresentasikan paragraf dan elemen <*gif*> merepresentasikan gambar. Data yang direpresentasikan tersebut harus diapit oleh tag pembuka <*element*> dan tag penutup <*/element*>. Seperti pada contoh berikut :

```
<img> .......isi elemen...........</img>
```

Elemen atau *tag* dalam XML harus ditulis dengan benar karena *case sensitive* dan setiap penulisan tag pembuka harus <> terlebih dahulu ditutup dengan *tag* penutup </> pasangannya sebelum dilakukan penulisan *tag* pembuka baru atau *tag* penutup yang bukan pasangannya

- XML Attribut & nilai

Atribut digunakan untuk menjelaskan tambahan informasi mengenai sebuah elemen, atau dengan kata lain merupakan bagian dari elemen yang merepresentasikan hal-hal yang dibentuk dari elemen tersebut. Misalkan untuk elemen <*img*> mempunyai atribut "*src*" yang merepresentasikan alamat gambar. Sedangkan Nilai adalah nilai dari atribut yang ada pada elemen tersebut misalkan menujukan alamat file "D:/Picture/wall.gif ", dan penulisan atribut harus dalam tanda kutip:

```
<img src="D:/Picture/wall.gif"></img>
```

- XML *Entity*

Entity adalah simbol yang merepresentasikan suatu informasi. Dengan menggunakan entity XML kita bisa menggantikan kalimat yang panjang atau satu blok elemen yang sering kita gunakan dengan sebuah pengenal singkat. Format entiti pada XML adalah tanda &, diikuti dengan nama dari simbol dan diakhiri dengan *semicolon*, seperti contoh berikut [9] :

Tabel 2.2. Contoh Entity

Entity	karakter	nama karakter
&lt;	<	less than
&gt;	>	greater than
&apos;	'	apostrophe

1.11 Tools Ekstraksi Data Web

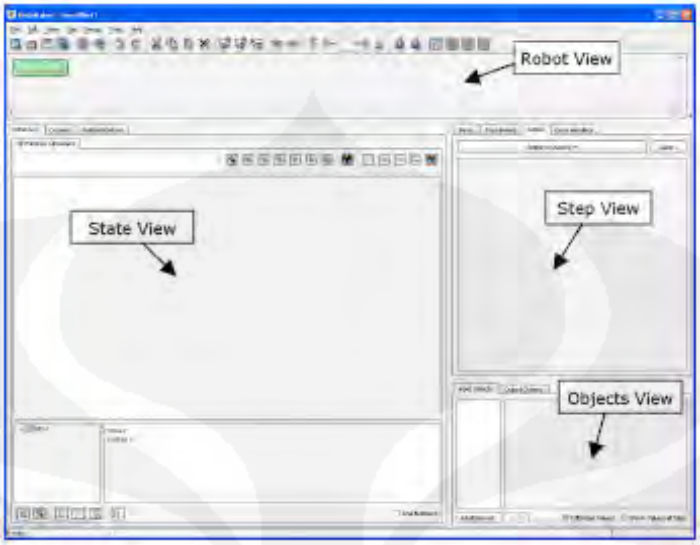
Sekarang ini terdapat beberapa *tools* yang dapat digunakan untuk proses ekstraksi data web baik *open source* ataupun berbayar. *Tool-tool* tersebut berfungsi untuk membantu membuat program ekstraksi data web dengan lingkungan pemrograman visual sehingga lebih mudah untuk digunakan karena dapat mengurangi penulisan skrip yang digunakan untuk ekstraksi data web. Beberapa *tools* tersebut antara lain:

2.5.1. Kapow Mashup Server 6.3 Robomaker

Kapow Mashup Server 6.3 Robomaker merupakan suatu *open service platform* dari openkapow, dimana pengguna dapat menggunakan suatu program tersendiri dan menjalankannya dari www.openkapow.com atau diinstalkan pada komputer masing-masing dengan gratis.

Konsep penting dari Kapow Mashup Server 6.3 Robomaker adalah robot, dimana robot ini adalah suatu program yang didesain untuk menjalankan suatu tugas tertentu, biasanya terkait *website*. Pada dasarnya robot dapat diprogram untuk secara otomatis semua yang dapat dilakukan pada sebuah *browser*.

Robomaker adalah lingkungan untuk pemrograman robot dalam sebuah bahasa pemrograman dengan kegunaan khusus dengan sintak dan semantik tersendiri. Lingkungan pemrograman dalam robomaker dapat terlihat seperti pada Gambar 2.6 berikut [12] :



Gambar 2.6. Tampilan Utama Jendela Robomaker

Robot *view* terletak di bawah ikon toolbar dari robomaker main window. Robot *view* ini akan menampilkan tahapan (*step*) dan koneksi dari *step* tersebut sehingga membentuk sebuah robot. *Step-step* yang ada pada robot view mempunyai beberapa elemen seperti nama, *list tag*, dan aksi atau akivitas yang dikerjakan step tersebut. Rangkaian tahapan dapat terlihat seperti berikut:

Gambar 2.7. Contoh Rangkaian Tahapan Robomaker

State *view* terleak dibawah robot *view* di sebelah kiri Jendela Utama Robomaker main window. Pada state view dapat terlihat keadaan robot aktual, atau tampilan dari yang sedang dilakukan robot, misalnya membuka salah satu halaman web. *State view* ini terdiri dari beberapa bagian diantaranya *tag path view, browser view, tree view* dan *source view* seperti Gambar 2.8 berikut :

Gambar 2.8. Tampilan Jendela Utama Page View Robomaker

Step view terletak di sebelah kanan *state view*. *Step view* ini memperlihatkan konfigurasi dari tahapan aktual. Pada *step view* ini juga dapat dilakukan konfigurasi untuk tahapan yang akan dibuat pada *Robot view*.

Objek *view* terletak di bawah *step view*. Objek *view* ini memperlihatkan objek dari keadaan tahapan aktual baik input ataupun output dari robot.

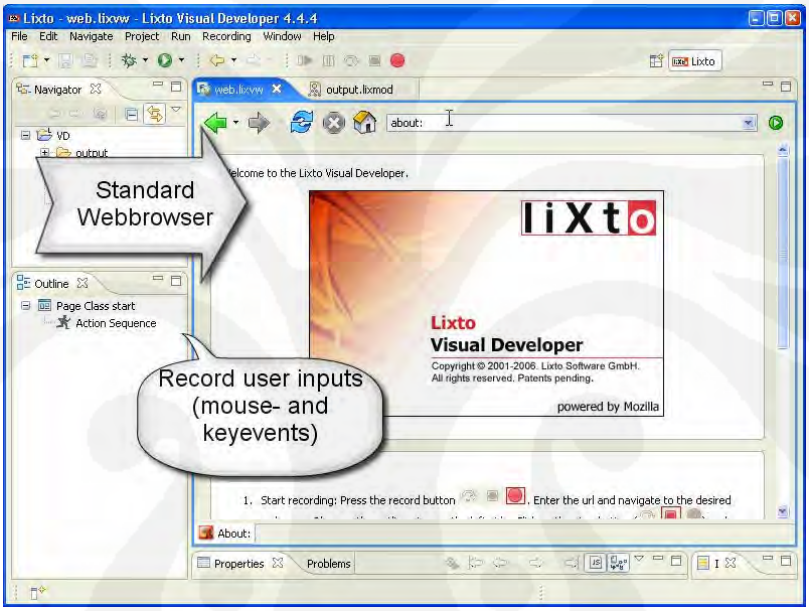
2.5.2. Lixto Visual Developer

Lixto Visual Developer (Lixto VD) adalah lingkungan pengembangan visual untuk program-program ekstraksi data web. Program yang dibuat dengan Lixto VD dapat disebarluaskan dan dieksekusi dengan Lixto server product. Lixto VD ini dibangun pada eclipse platform dan sebagai fitur dari Mozilla *browser* engine. Beberapa keuntungan yang ditawarkan oleh lixto antara lain :

- Lebih cepat dalam pengembangan karena rangkaian tool yang digunakan sudah standar dan memungkinkan untuk program web data ekstraksi yang interaktif;
- Lebih kuat dan dapat diandalkan dalam eksekusi program karena pengalamannya dalam algoritma pattern matching dan kualitas dalam mekanisme pengecekannya;

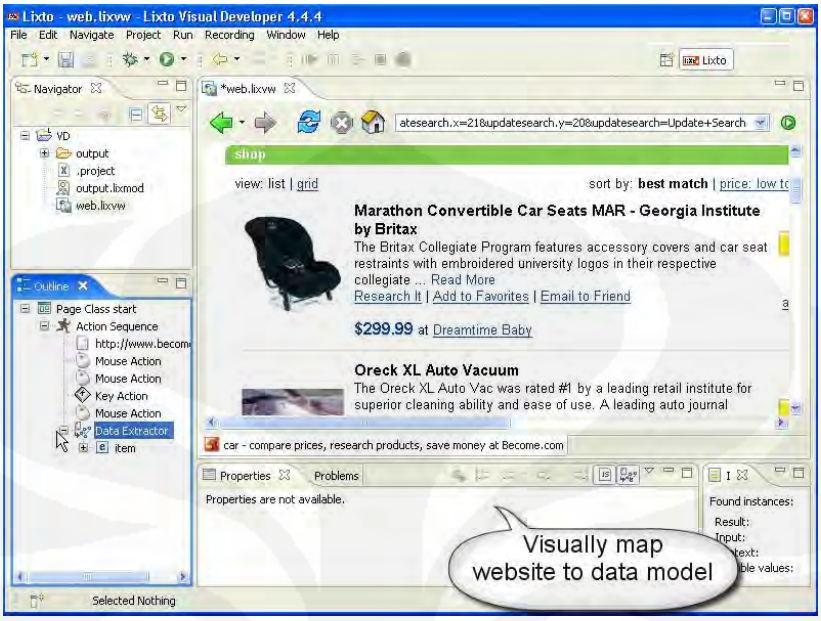
- Program lebih mudah dalam maintenance karena bahasa ekstraksi yang deklaratif.

Berikut gambaran tampilan Lixto VD, yang terdiri dari Standar Web browser yang berguna untuk menampilkan halaman web yang akan diekstraksi. *Record user inputs* yang berguna sebagai urutan dari proses yang dilakukan dalam ekstrasksi data web pada program.



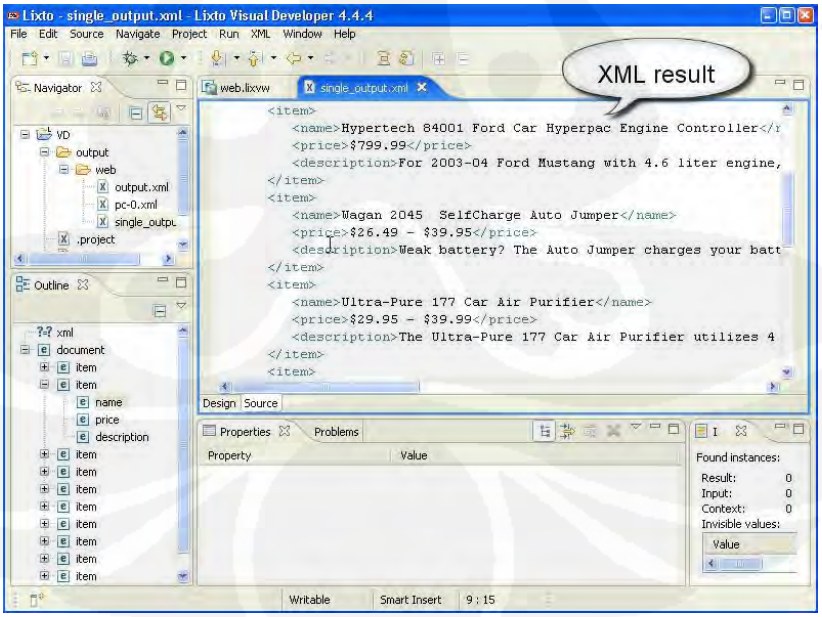
Gambar 2.9. Tampilan Standar Web Browser dan Record User Inpus Lixto

Bagian lainnya yang digunakan pada lingkungan pemrograman ini adalah *visually map website to data model*, yang berfungsi untuk melakukan pendefinisian pola atau *pattern* dan *filter* pada data yang diekstraksi. Seperti terlihat pada Gambar 2.10 berikut.



Gambar 2.10. Tampilan Visually Map Website To Data Model Lixto

Juga dapat dilihat *source* dari halama web yang sedang dibuka, seperti terlihat pada Gambar 2.11. berikut :



Gambar 2.11. Tampilan Source Dari Halaman Yang Dibuka Pada Lixto

Lixto ini menawarkan beberapa fitur seperti pembacaan halaman web sesuai standar, definisi Pola dan *filter* dari ekstraksi data web yang interaktif; penandaan dari informasi yang diekstraksi dari halaman web, cocok untuk visual debug selama proses pngembangan, dan Umpan balik visual dari hasil perubahan program ekstraksi data web diterima langsung. Selain itu pada proses ekstraksinya Lixto mempunyai Mempunyai Xpath 2 yang dapat berguna untuk memanfaatkan struktur dari dokumen HTML dan mengidentifikasi elemen dokumen, tidak adanya batasan struktur model data, dan mekanisme pengecekan yang menjamin kualitas data yang diekstraksi.

1.12 Portable Document Format

Portable Document Format (PDF) adalah suatu format file yang digunakan untuk pertukaran dokumen digital, yang dibuat pada tahun 1993 oleh Adobe system. PDF ini digunakan untuk merepresentasikan dokumen dua dimensi pada aplikasi perangkat lunak, *hardware* ataupun sistem operasi yang independen, dimana file PDF ini terdiri dari kumpulan deskripsi untuk *layout* dua dimensi seperti text, jenis huruf, citra, dan vektor grafik dua dimensi [17].

Awalnya penggunaan PDF tidak terlalu banyak karena perangkat lunak untuk membuat dan membacanya tidak tersedia secara gratis dan tidak mendukung untuk hiperlink ekstrenal. Ukurannya yang besar juga menjadikan PDF ini kurang populer. Pada saat itu juga PDF harus bersaing dalam tingkat penggunaannya dengan format lain seperti *Envoy, Common Ground Digital Paper,* dan *PostScript* (.ps). *PostScript* adalah format yang juga diciptakan oleh Adobe dan sebagian fungsinya diimplementasikan pada PDF. Laju peningkatan penggunaan dokumen PDF meningkat dengan pesat setelah Adobe mulai mendistribusikan perangkat lunak *Acrobat Reader* secara gratis dan membebaskan pembuatan aplikasi pembuat maupun pembaca dokumen PDF tanpa perlu membayar royalti kepada *Adobe System* selaku pemegang hak paten PDF. Beberapa versi PDF sejak pertamakali dibuata antara lain :

- Tahun 1993 – PDF 1.0 atau Acrobat 1.0.
- Tahun 1994 – PDF 1.1 atau Acrobat 2.0 dengan fitur *Passwords, device-independent color, threads and links.*

- Tahun 1996 – PDF 1.2 atau Acrobat 3.0 dengan fitur *Interactive page elements, mouse events, multimedia types, Unicode, advanced color features and image proxying.*
- Tahun 1999 – PDF 1.3 atau Acrobat 4.0 dengan fitur *Digital signatures; ICC and DeviceN color spaces; JavaScript actions.*
- Tahun 2001 – PDF 1.4 atau Acrobat 5.0 dengan fitur JBIG2; *transparency; OCR text layer.*
- Tahun 2003 – PDF 1.5 atau Acrobat 6.0 dengan fitur JPEG2000; *linked multimedia.*
- Tahun 2005 – PDF 1.6 atau Acrobat 7.0 dengan fitur *Embedded multimedia; XML forms; AES encryption.*
- Tahun 2006 – PDF 1.7 atau Acrobat 8.0.
- Tahun 2008 – PDF 1.7, Adobe Extension Level 3 / Acrobat 9.0.

Pada dasarnya PDF mengkombinasikan tiga teknologi yaitu :

- Sub-set dari pemrograman deskripsi halaman *PostScript* untuk menghasilkan tampilan dan grafik.
- Sistem penempatan/pemindahan huruf untuk mengijinkan perpindahan huruf di dalam dokumen.
- Sistem penyimpanan terstruktur untuk menempatkan dan mengkompresi elemen-elemen dokumen ke dalam satu berkas.

Pada 1 Juli 2008, ISO menerbitkan standar untuk PDF dengan ISO 32000 - 1:2008 PDF dengan judul *Document management -- Portable document format -- Part 1: PDF 1.7*. dan beberapa pdf telah digunakan untuk beberapa aplikasi ISO seperti :

- PDF/X untuk pencetakan dan grafik pada ISO 15930.
- PDF/A untuk arsip perusahaan/pemerintah/perpustakaan/ lingkungan seperti pada ISO 19005.
- PDF/E untuk pertukaran gambar teknik.
- PDF/UA untuk aksesibilitas universal.

File PDF terdiri dari beberapa objek antara lain :

- *Boolean values*, merepresentasikan nilai true atau false.
- *Numbers.*
- *Strings*.
- *Names.*
- *Arrays*, terdiri dari beberapa objek.
- *Dictionaries*, kumpulan dari ojke-objek yang diindeks dengan nama.
- *Streams*, mengandung sejumlah besar data.
- *The Null object.*

Objek pada pdf ini bisa secara langsung ataupun tidak ditanamkan objek lainnya, objek tidak langsung akan dinomori dengan *objek number*. Pada pdf ini terdapat sebuah tabel indek atau disebut *xref table* yang akan merepresentasikan pergeseran *byte* untuk semua objek tidak langsung yang dihitung dari awal file. Dengan demikian menjadi lebih efisien untuk mengakses objek secara acak pada file dan tidak perlu menulis ulang semua file jika ada perubahan. Sejak pdf versi 1.5, objek tidak langsung ditempatkan di *stream* khusus yang disebut objek *stream* sehingga dapat mengurangi ukuran file.

Pdf file mempunyai dua *layout* yaitu *not linear* atau *not optimized* yang menggunakan ruang lebih kecil dari *linear*, akan tetapi lebih lambat untuk diakses karena membutuhkan waktu untuk merakit kembali halaman yang terpecah-pecah. *Linear* atau *optimized* dibentuk dengan cara yang memungkinkan untuk terbaca pada *web browser*. Teks pada pdf ini direpresentasi dengan elemen teks yang berada pada stream. Sebuah teks elemen menunjukkan dimana suatu karakter akan diletakkan.

BAB 3
PERANCANGAN

1.13 Spesifikasi dan fungsi sistem

Perancangan merupakan tahapan penting dalam pembuatan suatu sistem, sehingga dengan perancangan yang baik diharapkan akan dihasilkan suatu sistem yang sesuai dengan fungsi dan tujuan dari dibuatnya sistem tersebut. Pada perancangan ini umumnya terdiri dari beberapa tahapan seperti menentukan spesifikasi dan fungsi sistem, menentukan cara kerja sistem, mengidentifikasi hal-hal yang dibutuhkan sistem dan menentukan *tool* atau alat bantu yang digunakan untuk pembuatan sistem.

Sistem yang dirancang merupakan sistem berbasis web yang berfungsi untuk mencari dan menampilkan indeks sitasi dari jurnal-jurnal yang bersumber dari institusi-institusi yang ada di Indonesia. Jurnal yang diproses oleh sistem adalah jurnal yang dapat diakses bebas dan dalam format file pdf. Sistem menampilkan indeks sitasi dengan menggunakan suatu antarmuka pemakai yang berbentuk halaman web yang dapat diakses bebas memalui Internet.

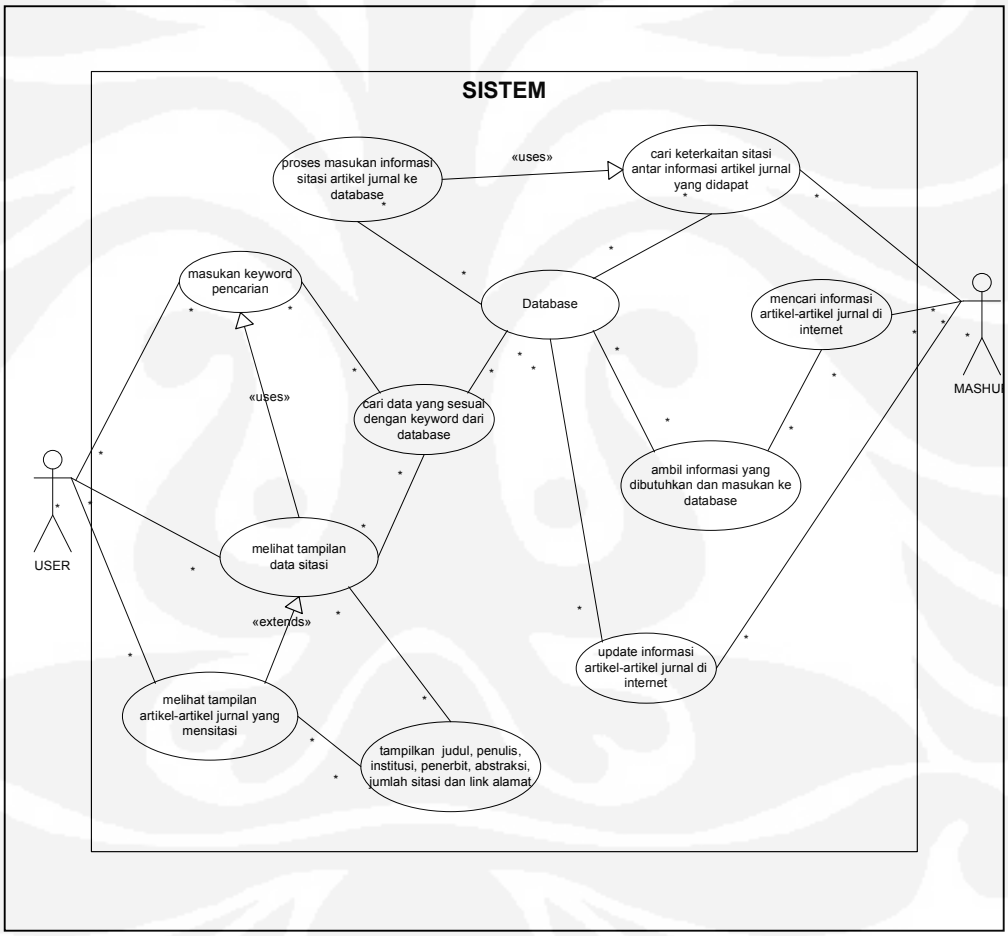
1.14 Cara kerja sistem

Sistem bekerja dengan cara mengumpulkan jurnal -jurnal yang didapat dari institusi-institusi yang ada di indonesia, kemudian menyimpan data-data yang diperlukan ke suatu *database*, lalu mengolah data jurnal tersebut untuk mengidentifikasi keterkaitan dengan jurnal lainnya sehingga dengan proses tersebut dihasilkan suatu indeks sitasi jurnal.

Dari sisi pengguna, sistem akan mirip seperti *search engine*, dimana pengguna akan memasukan suatu kata kunci berdasarkan judul ataupun penulis pada *field* yang ada. Setelah perintah pencarian dieksekusi oleh pengguna, maka sistem akan mencari data sesuai kata kunci tersebut pada *database* lalu menampilkannya kepada pengguna sebagai hasil pencarian. Selain itu sistem dapat melakukan pembaharuan atau *update* data jurnal yang ada dengan cara melakukan pengecekan terhadap perubahan atau penambahan data jurnal pada sumber-sumber jurnal yang ditentukan.

1.15 Mengidentifikasi kebutuhan sistem

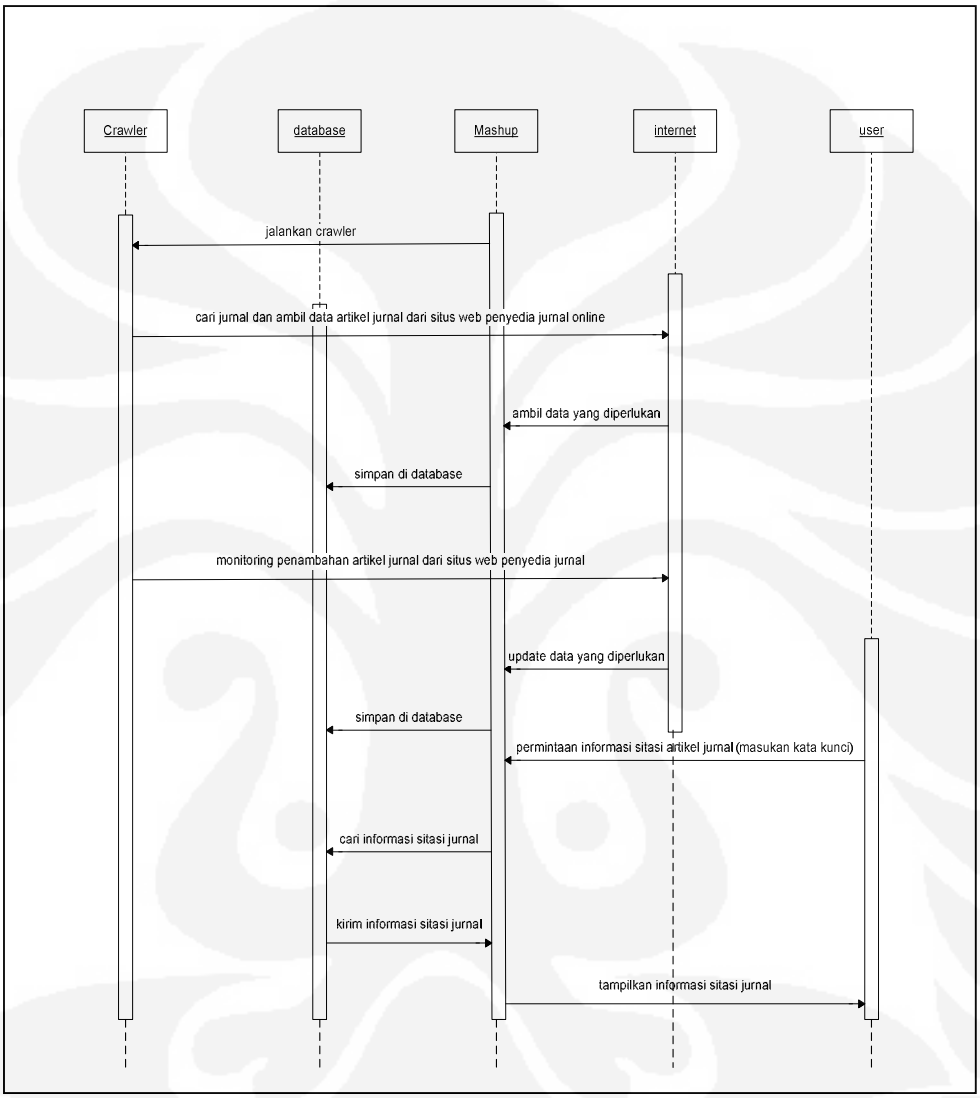
Sebelum melangkah pada tahapan lainnya, terlebih dahulu harus ditentukan hal-hal yang akan menjadi kebutuhan sistem yang akan dibuat, sehingga dengan mengetahui kebutuhan sistem secara jelas akan lebih memudahkan dalam tahapan perancangan selanjutnya juga mempermudah dalam tahapan pembuatan sistem. Untuk menggambarkan kebutuhan sistem dilihat dari interaksi pengguna dengan sistem dapat dituangkan dalam suatu diagram *use-case* seperti pada Gambar 3.1 di bawah ini :



Gambar 3.1. Use Case Diagram Sistem

Selain itu, untuk mengetahui bagaimana urutan komunikasi yang terjadi antara komponen yang satu dengan lainnya yang terdapat pada sistem yang akan

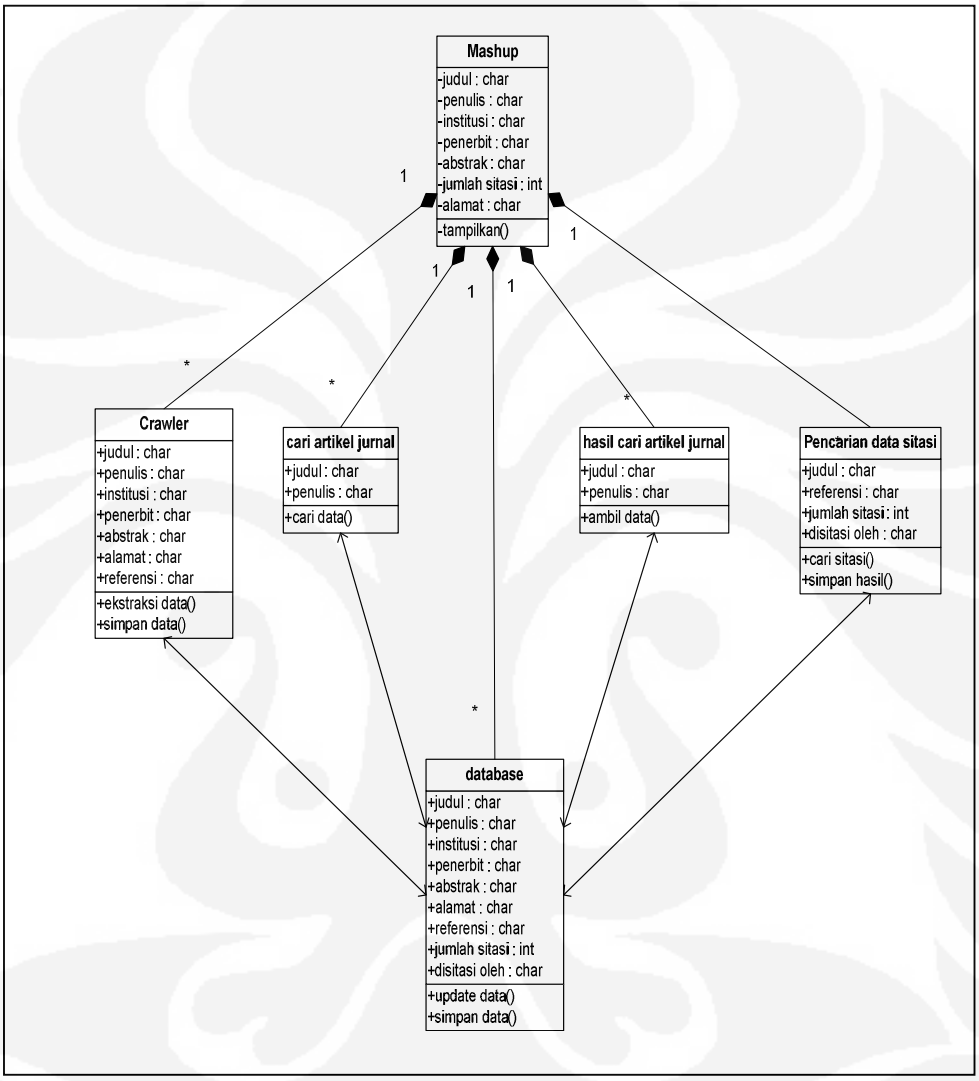
dibuat, dapat digambarkan dengan menggunakan *sequence diagram* seperti terlihat pada Gambar 3.2 berikut :



Gambar 3.2. Diagram Sequence Dari Sistem

Pada saat sistem berjalan, selain data yang disimpan di *database*, data tersebut juga akan diproses oleh masing-masing komponen terkait yang ada pada sistem. Masing-masing komponen yang ada pada sistem tersebut akan memerlukan suatu data atau variabel untuk diproses sehingga menghasilkan keluaran hasil proses atau data akhir yang dihasilkan sesuai dengan yang

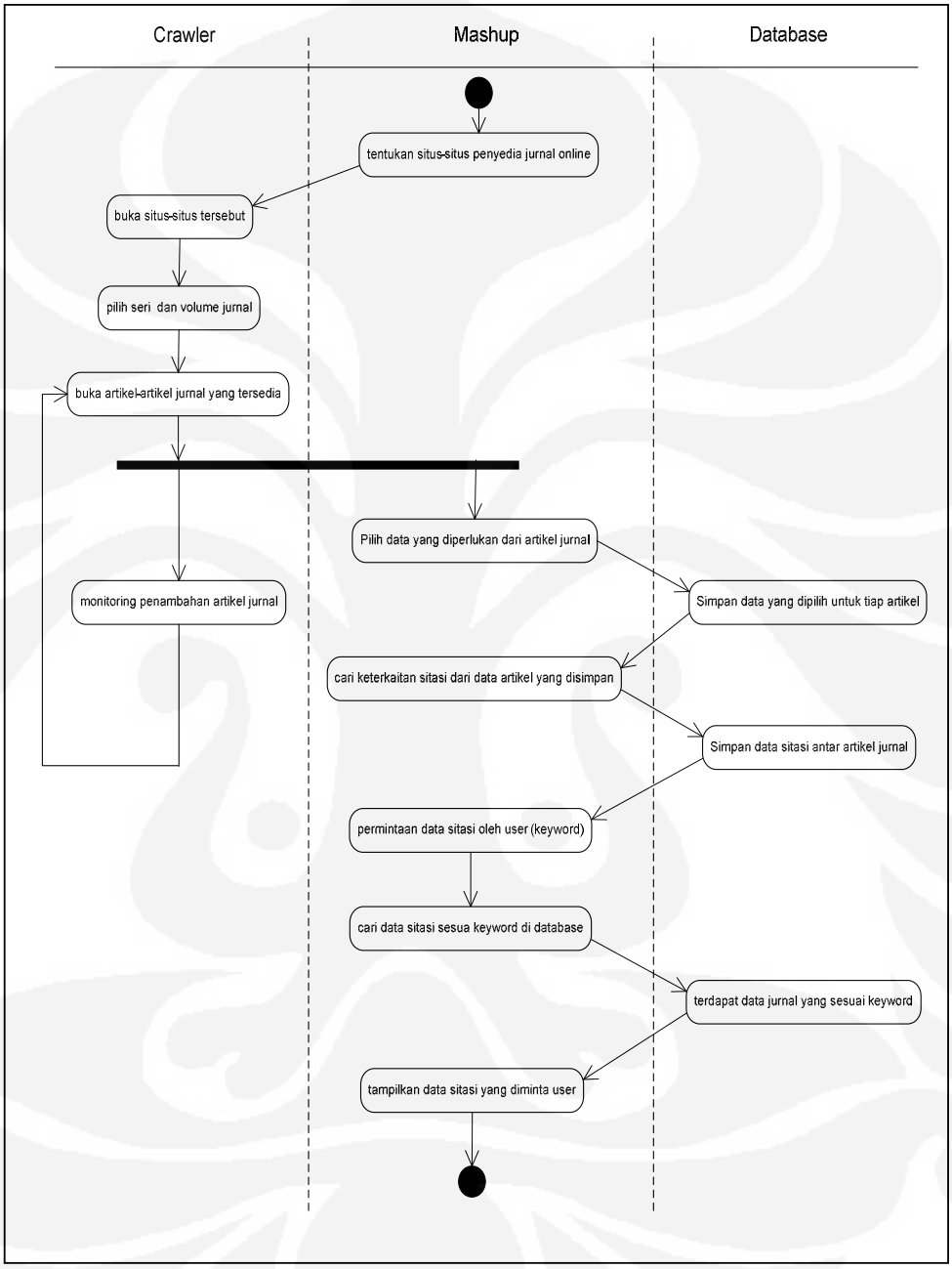
diharapkan yaitu terbentuknya suatu indeks sitasi. Untuk mengetahui variabel-variabel atau data-data yang dibutuhkan tersebut dapat terlihat pada suatu *class diagram* seperti pada Gambar 3.3 berikut :



Gambar 3.3. Class Diagram sistem

Untuk memperjelas langkah-langkah atau alur kerja dari keseluruhan sistem yang akan dibuat, juga untuk mengetahui operasi-operasi yang dilakukan oleh masing-masing komponen pada sistem. Digambarkan dengan menggunakan

salah satu diagram yang ada pada UML yaitu dengan menggunakan suatu *activity diagram* seperti pada Gambar 3.4 berikut ini :



Gambar 3.4. Activity Diagram Sistem

Tahapan selanjutnya adalah perancangan halaman web yang akan digunakan sebagai antar muka bagi pengguna, halaman web yang dibuat diharapkan memenuhi kriteria berikut :

- Tampilan / *layout* menarik
- Fungsi-fungsi yang ada mudah dipahami dan digunakan sehingga memperkecil tingkat kesalahan pemakaian web oleh pengguna.
- Maksud dari isi yang ditampilkan jelas dan mudah dimengerti juga sesuai dengan tujuan awal pembuatan sistem.

Halaman web yang akan dibuat terdiri dari dua halaman yaitu:

1. Halaman pembuka yang merupakan halaman yang pertama kali muncul saat pengguna masuk ke situs ini. Halaman ini terdiri dari :
   - Nama situs
   - Kotak teks untuk mengisikan kata kunci pencarian jurnal.
   - Tombol pilihan untuk pemilihan tipe kata kunci (judul atau penulis).
   - Tombol untuk memulai pencarian.
   - Tulisan singkat mengenai isi dan fungsi situs.
2. Halaman utama yang merupakan halaman yang digunakan untuk menampilkan hasil pencarian dan melakukan pencarian-pencarian selanjutnya. Halaman ini akan terdiri dari tiga bagian utama dengan fungsi dan spesifikasi isi sebagai berikut :
   - Kepala halaman

     Kepala halaman ini terdiri dari :
     - Judul / nama web.
     - Kotak teks untuk mengisikan kata kunci pencarian jurnal.
     - Tombol pilihan untuk pemilihan tipe kata kunci (judul atau penulis).
     - Tombol untuk memulai pencarian.
     - *Link* untuk melihat halaman berikutnya.
     - *Link* untuk melihat halaman sebelumnya.
     - Teks untuk menampilkan hasil jumlah pencarian.
     - Teks untuk menampilkan urutan jumlah hasil pencarian yang ditampilkan.

- Badan Halaman

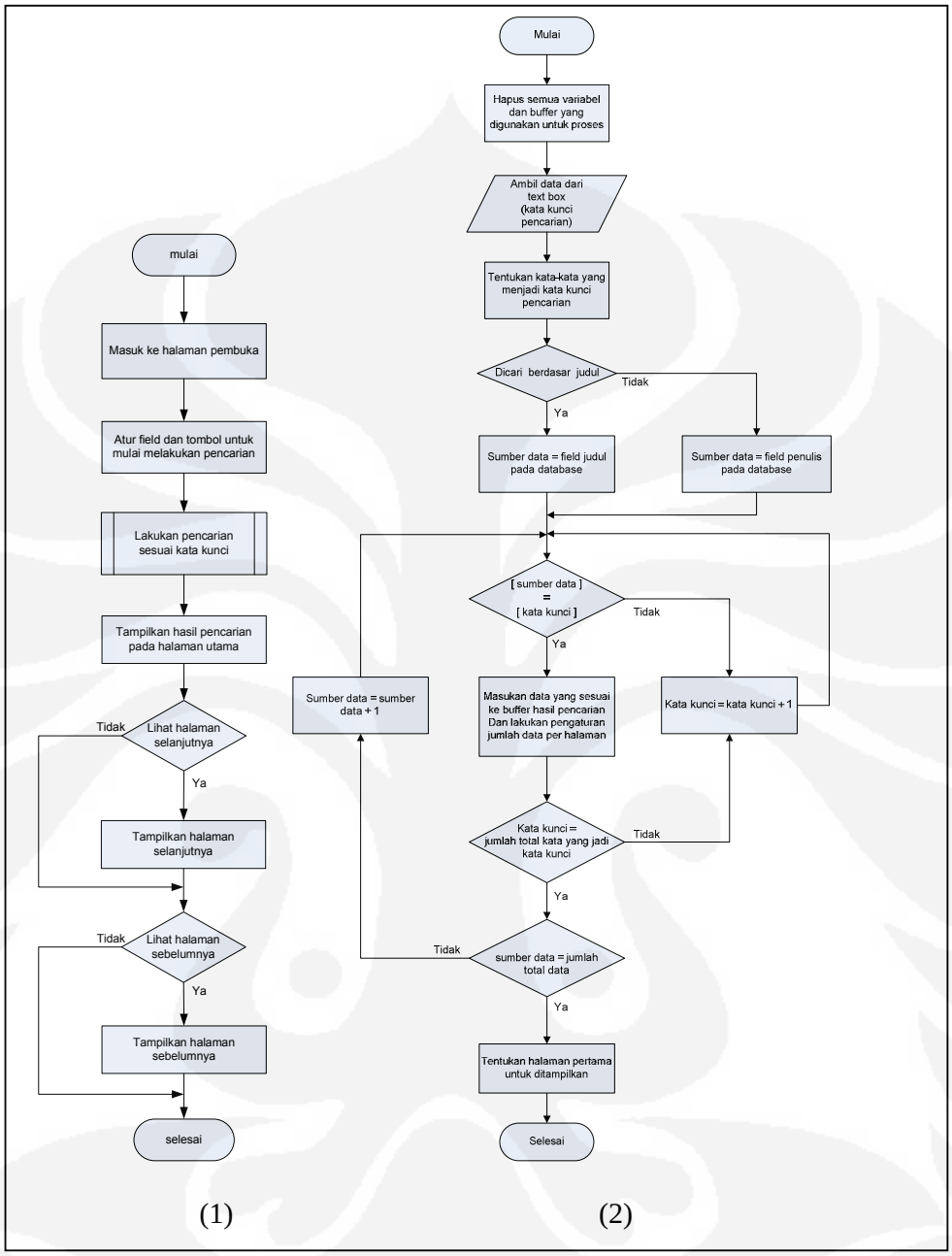
  Badan halaman ini berisi data-data hasil pencarian yang terdiri dari :

  - Judul artikel jurnal (beserta *link* ke halaman sebenarnya).
  - Penulis.
  - Penerbit.
  - Institusi penulis.
  - Potongan abstraksi (maksimal 250 kata).
  - Jumlah sitasi terhadap judul jurnal diatas.
  - Jumlah data jurnal yang ditampilkan per halaman adalah 10 data artikel.

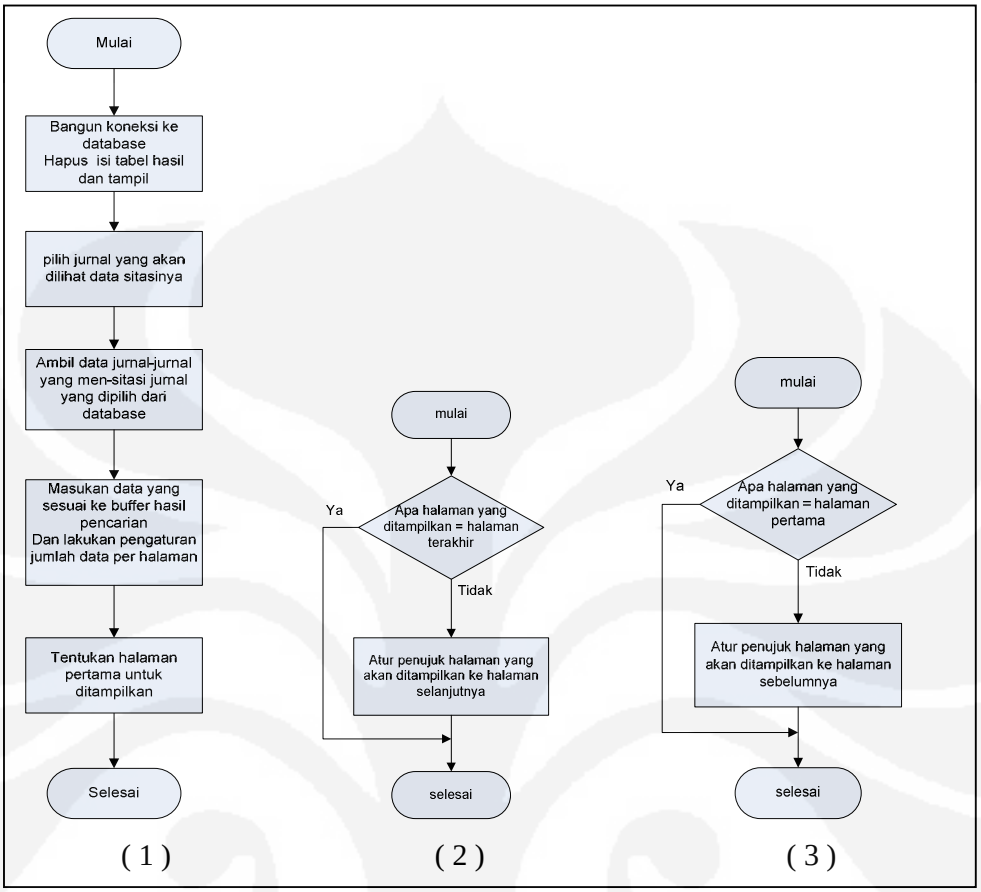
- Kaki halaman

  Berupa grafik atau tulisan pelengkap identitas web.

Setelah halaman web dan fungsi-fungsi yang terdapat di dalamnya terdefinisi dengan jelas, maka untuk menggambarkan alur kerja dari fungsi-fungsi yang akan dibuat pada halaman web tersebut digunakan diagram alir seperti terlihat pada gambar 3.6 berikut :



Gambar 3.5. Diagram alir main program halaman web (1), dan Diagram alir pencarian data jurnal sesuai kata kunci (2).



Gambar 3.6. Diagram alir untuk melihat jurnal yang mensitasi (1), Diagram alir untuk melihat halaman selanjutnya (2), dan Diagram alir untuk melihat halaman sebelumnya (3)

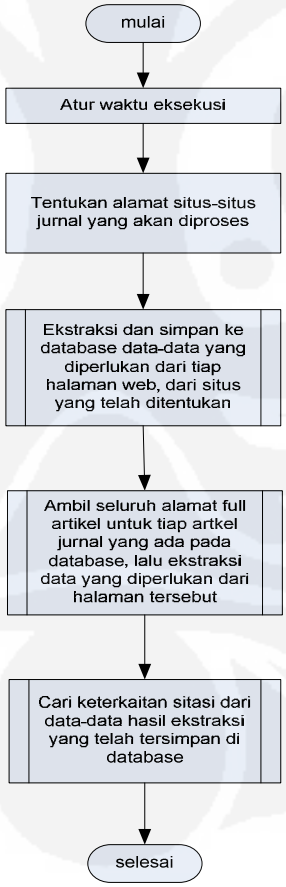
Setelah semua perancangan yang terkait dengan cara kerja sistem secara keseluruhan didefinisikan di atas, baik itu mengenai interaksi antara pengguna dengan sistem, urutan komunikasi antar komponen yang ada pada sistem, variabel dan data yang digunakan pada sistem dan juga alur atau langkah kerja dari sistem (sesuai dengan fungsi masing-masing komponen yang ada pada sistem). Halaman web yang akan digunakan sebagai antarmuka dengan pengguna beserta masing-masing fungsinya. Selanjutnya pembuatan sistem dengan melakukan *coding* untuk fungsi-fungsi yang terdapat pada masing-masing komponen sehingga keseluruhan sistem dapat terealisasi seluruhnya, dan dapat bekerja dan dipergunakan sesuai dengan yang diharapkan dan tujuan awal pembuatan sistem. Untuk pembuatannya digunakan bahasa PHP dan MySQL sebagai *database*.

BAB 4
IMPLEMENTASI

Pada implementasi aplikasi ekstraksi web untuk keperluan indeks sitasi ini, sistem bekerja dengan mengekstraksi halaman-halaman web penyedia jurnal dengan situs-situs berikut :

- http://journal.ui.ac.id (Universitas Indonesia)
- http://proceedings.itb.ac.id (Institut Teknologi Bandung)
- http://ejournal.unud.ac.id (Universitas Udayana)
- http://puslit2.petra.ac.id/ejournal (Universitas PETRA)

dimana proses yang dilakukan oleh sistem untuk mendapatkan data-data yang diperlukan untuk keperluan indeks sitasi ini dapat terlihat pada diagram alir pada Gambar 4.1 berikut :



Gambar 4.1 . Diagram Alir Proses Kerja Sistem Ekstraksi

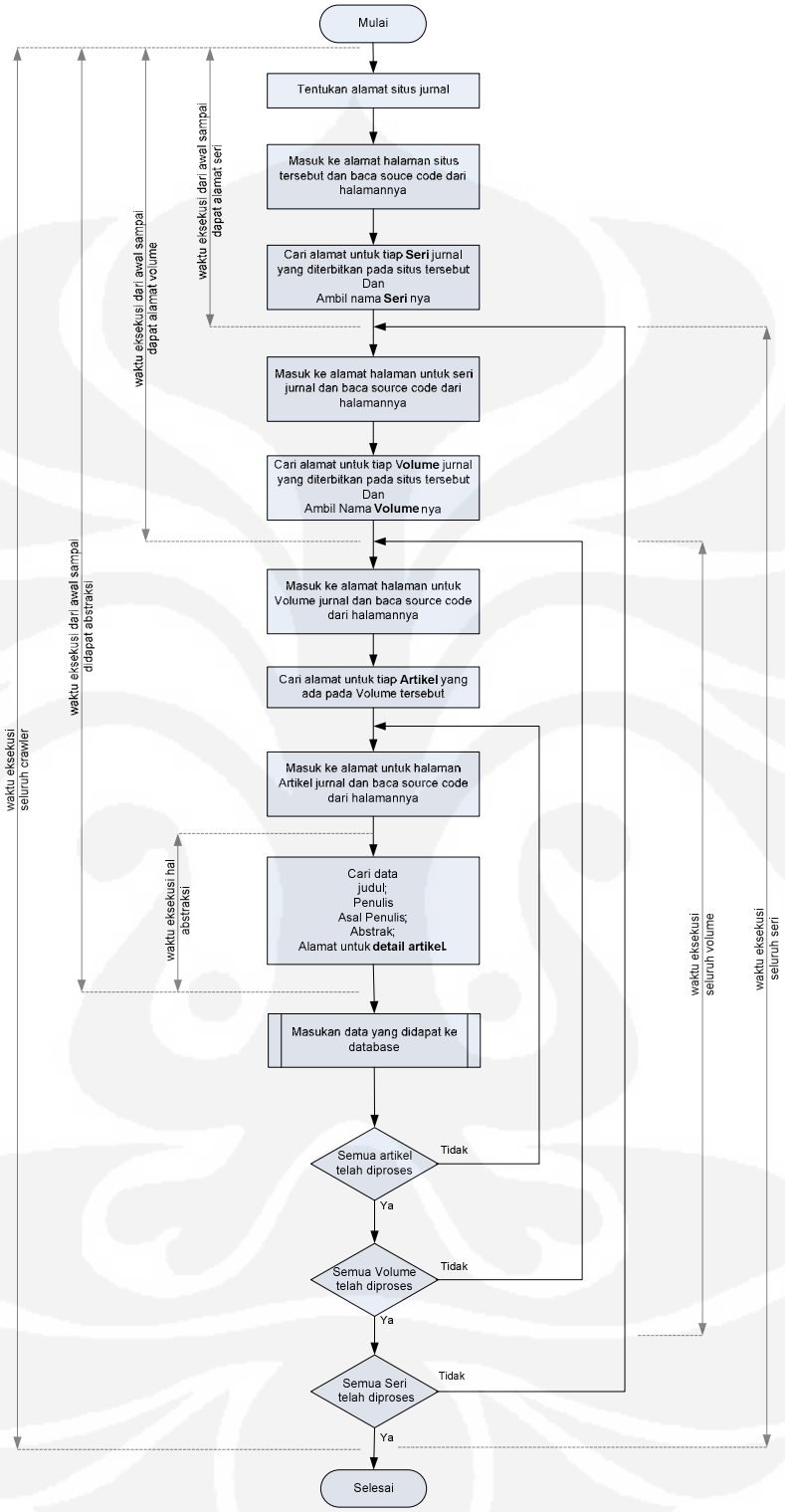
1.16 Pengujian Kinerja Sistem Ekstraksi

Pengujian sistem dilakukan dengan cara menjalankan program yang telah dibuat. Dari hasil pengujian ini dapat diketahui data-data yang berhasil diekstraksi dan juga dapat diketahui beberapa informasi berupa waktu dan penggunaan memori terkait kinerja sistem. Untuk mendapatkan informasi tersebut digunakan bantuan fungsi yang disediakan oleh PHP, fungsi tersebut disisipkan atau disimpan dalam skrip program sehingga akan mengoptimalkan pengukuran informasi yang dibutuhkan.

Untuk mendapatkan informasi waktu digunakan fungsi `microtime()`, fungsi microtime ini berfungsi untuk menghasilkan nilai waktu saat ini dalam dua bagian yaitu detik dan mikrodetik yang dihitung dari 1 januari 1970. Fungsi yang digunakan untuk mengetahui besarnya memori terpakai saat skrip dijalankan, menggunakan fungsi `memory_get_usage()` yang menghasilkan output dalam satuan *Byte*. Kedua fungsi tersebut akan menghasilkan informasi yang diperlukan saat fungsi tersebut turut tereksekusi dalam program aplikasi yang dibuat, dengan cara menambahkan perintah `echo()`, sehingga hasil dari fungsi ini akan ditampilkan pada halaman *browser* dan informasi yang diperlukan dapat diproses.

Pada pengujian aplikasi fungsi ekstraksi halaman web, yang berfungsi untuk melakukan kestraksi data-data yang diperlukan dari halaman web situs penyedia jurnal, dilakukan pengukuran waktu eksekusi dari beberapa titik aplikasi program. Pengukuran ini dilakukan dengan menyisipkan fungsi `microtime()` pada beberapa titik aplikasi program, seperti terlihat pada diagram alir cara kerja fungsi ekstraksi halaman web pada Gambar 4.2.

Pengukuran dilakukan beberapa kali untuk setiap situs penyedia jurnal, sehingga data yang didapat merupakan data hasil rata-rata dari beberapa kali proses pengukuran yang telah dilakukan. Pengukuran dilakukan dengan kecepatan transfer data maksimal 144 Kbps. Dengan dilakukannya pengkuran tersebut dapat diketahui besarnya waktu yang dibutuhkan untuk mengekstraksi data-data yang diperlukan dari tiap situs penyedia jurnal. Waktu eksekusi yang dibutuhkan, diukur dan dihitung dalam satuan detik, dengan ketelitian mikrodetik, sehingga jika terdapat perbedaan besarnya waktu eksekusi akan terlihat dengan jelas.



Gambar 4.2 . Diagram Alir Fungsi Ekstraksi Halaman Web

Eksekusi aplikasi ekstraksi halaman web sesuai diagram alir Gambar 4.2, akan menghasilkan informasi atau data yang berhasil diekstraksi dari situs penyedia jurnal, dan juga informasi tambahan terkait eksekusi aplikasi program tersebut. yang kemudian akan ditampilkan pada halaman *browser*, seperti terlihat pada Gambar 4.3.

Gambar 4.3 . Hasil Tampilan Aplikasi Program Ekstraksi Halaman Web

Jumlah data artikel jurnal yang berhasil diekstraksi dari proses eksekusi aplikasi ekstraksi halaman web adalah sebanyak 3358 dengan rincian pada Tabel 4.1 berikut :

Tabel.4.1. Sumber Dan Jumlah Artikel Jurnal Yang Berhasil Diekstraksi

Sumber	***jumlah artikel hasil ekstraksi***	***jumlah artikel aktual***
Universitas Indonesia	370	407
Institut teknologi bandung	183	183
Universitas kristen Petra	1518	1520
Universitas Udayana	1287	1296

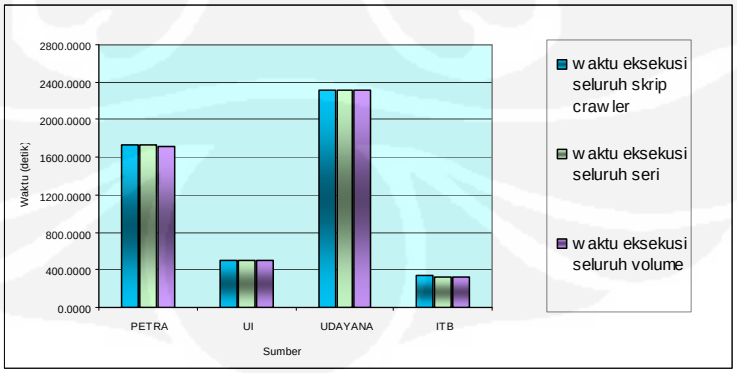
`

Informasi tambahan waktu eksekusi dari beberapa titik aplikasi program yang ditampilkan pada halaman *browser*, yang merupakan nilai rata-rata dari 10 kali pengukuran (data lengkap di Lampiran A) dapat dilihat pada Tabel 4.2 berikut:

Tabel 4.2. Waktu Eksekusi untuk Aplikasi Fungsi Ekstraksi Halaman Web

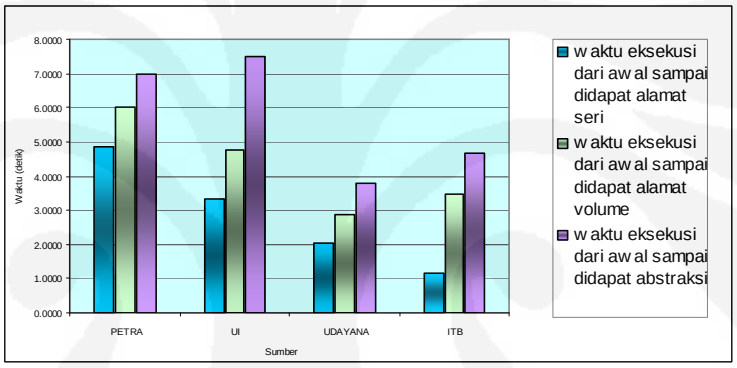
Parameter	***PETRA (detik)***	***UI (detik)***	***UDAYANA (detik)***	***ITB (detik)***
waktu eksekusi seluruh skrip crawler	1731.561837	504.5820959	2314.38659	332.0527513
waktu eksekusi seluruh seri	1726.081298	502.9297843	2312.437495	330.7665707
waktu eksekusi seluruh volume	1721.175804	502.1483421	2311.597343	327.9454458
waktu eksekusi dari awal sampai dapat alamat seri	4.875258467	3.34780097	2.014105304	1.178149147
waktu eksekusi dari awal sampai dapat alamat volume	6.019417117	4.75399423	2.862114613	3.483318293
waktu eksekusi dari awal sampai dapat abstraksi	7.002499753	7.494512492	3.79775282	4.676582253
waktu eksekusi hal abstraksi sampel 1	0.003169311	0.000773031	0.000413281	0.002579314
waktu eksekusi hal abstraksi sampel 2	0.008988754	0.000649433	0.000341244	0.003647535
waktu eksekusi hal abstraksi sampel 3	0.002750103	0.000847699	0.000379542	0.002350281
waktu eksekusi hal abstraksi sampel 4	0.002090243	0.000657074	0.000346335	0.002223288
waktu eksekusi hal abstraksi sampel 5	0.003402609	0.000659735	0.000363041	0.003951211

Data yang terdapat pada Tabel 4.2 dapat direpresetasikan dalam beberapa tampilan grafik seperti pada beberapa gambar grafik berikut ini :



Gambar 4.4. Grafik waktu Eksekusi Skrip Ekstraksi Halaman Web Keseluruhan Skrip Crawler, Keseluruhan Seri dan Keseluruhan Volume

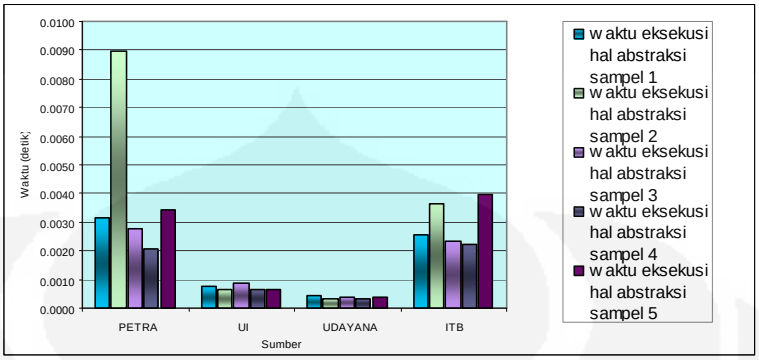
Dari grafik Gambar 4.4 terlihat perbedaan waktu eksekusi untuk tiap situs penyedia jurnal. Perbedaan waktu disebabkan adanya perbedaan jumlah seri, jumlah volume dan jumlah artikel jurnal untuk masing-masing volume. Kombinasi perbedaan antara jumlah seri dan jumlah volume untuk jumlah keseluruhan artikel jurnal yang sama, dapat mengakibatkan perbedaan waktu eksekusi karena akan terjadi perbedaan jumlah halaman web yang dibuka. Hal ini terlihat pada waktu yang dibutuhkan untuk mengekstraksi situs penyedia jurnal Universitas Udayana lebih lama dibanding Universitas Kristen Petra, meskipun dengan jumlah artikel yang sebaliknya, ini disebabkan kombinasi jumlah seri dan volume dari Universitas Udayana lebih banyak dibanding Universitas Kristen Petra.



Gambar 4.5. Grafik Waktu Eksekusi Skrip Ekstraksi Halaman Web Dari Awal Sampai Didapatkan Halaman Abstraksi Artikel Jurnal Pertama Kali

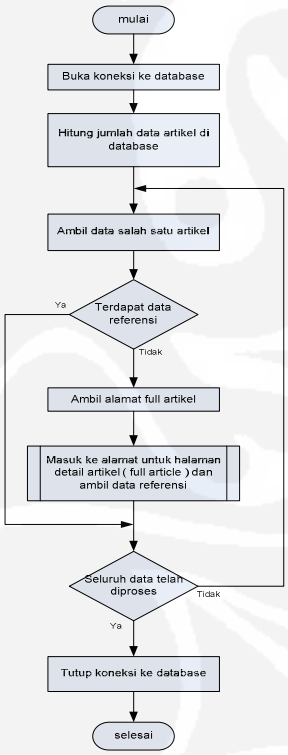
Dari Gambar 4.5 besarnya waktu eksekusi untuk mendapatkan alamat seri, alamat volume dan alamat abstraksi pertama kali terdapat perbedaan, hal ini disebabkan karena jumlah halaman web yang harus dibuka untuk mendapatkan masing-masing data yang diperlukan di atas, berbeda-beda untuk tiap situs. Akan tetapi selain disebabkan perbedaan jumlah halaman web yang harus dibuka, hal ini disebabkan juga oleh lamanya waktu yang dibutuhkan untuk membuka halaman web tersebut.

Sedangkan perbedaan besarnya waktu eksekusi untuk melakukan pemisahan judul, penulis, institusi, penerbit, alamat artikel lengkap, dan abstraksi seperti terlihat pada grafik Gambar 4.6 berikut, disebabkan oleh perbedaan format penulisan tampilan halaman abstraksi web dari artikel jurnal yang diproses.



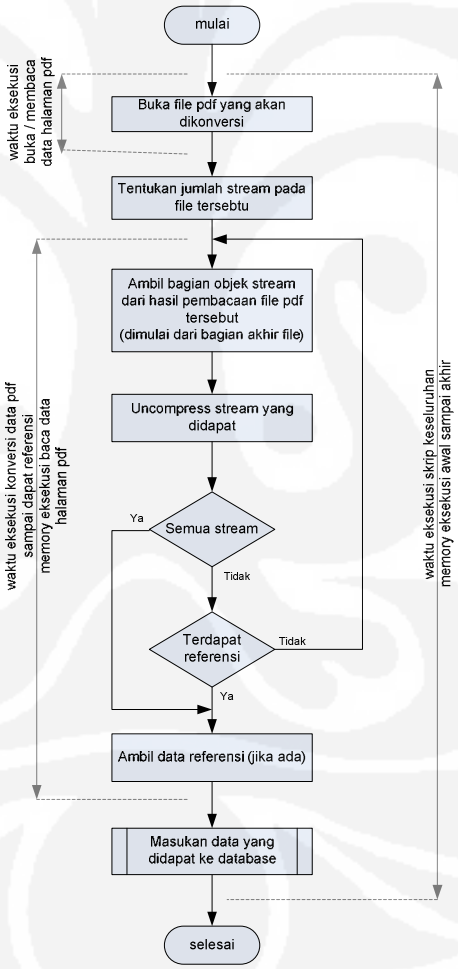
Gambar 4.6. Grafik Waktu Eksekusi Skrip Ekstraksi Halaman Web untuk Mendapatkan Informasi Judul, Penulis, Institusi, dan Abstraksi dari Halaman Abstraksi Artikel Jurnal

Aplikasi Fungsi lainnya yang ada dalam sistem adalah adalah mendapatkan data referensi dari artikel jurnal yang alamatnya telah disimpan di database yang dihasilkan dari proses aplikasi fungsi ekstraksi halaman web yang sebelumnya telah terlebih dahulu dieksekusi. Prosesnya dapat dilihat dari diagram alir Gambar 4.7.



Gambar 4.7 . Diagram Alir Fungsi Mendapatkan Data Referensi

Pada fungsi mendapatkan data referensi terdapat fungsi untuk ekstraksi halaman pdf, yang berguna untuk mendapatkan data bagian referensi dari artikel jurnal lengkap. Karena data yang diproses berbentuk file pdf maka diperlukan proses untuk mengkonversi isi file pdf agar dapat terbaca dengan baik, dan dapat diambil data bagian referensinya secara otomatis. Informasi lainnya yang diambil dari hasil eksekusi fungsi ini adalah waktu eksekusi dan besarnya penggunaan memori saat eksekusi aplikasi ekstraksi halaman pdf dilakukan. Pengukuran waktu dan memori ini dilakukan dengan menyisipkan fungsi `microtime()` dan fungsi `memory_get_usage()` pada beberapa titik aplikasi program, seperti terlihat pada diagram alir cara kerja fungsi ekstraksi Pdf pada Gambar 4.8. di bawah ini :



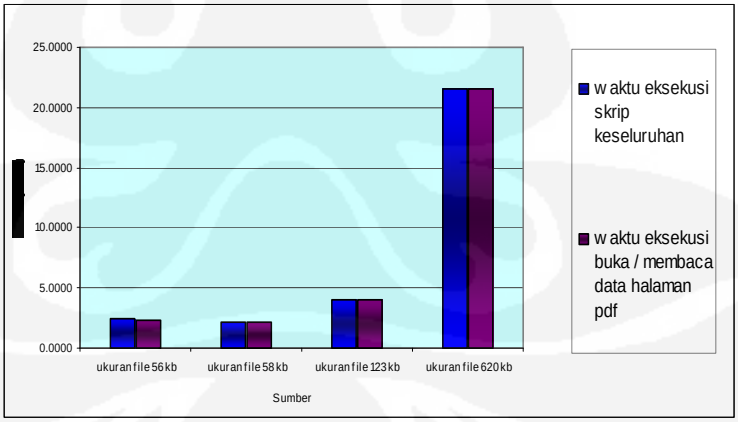
Gambar 4.8 . Diagram Alir Fungsi Ekstraksi Pdf

Pengukuran dilakukan 10 kali terhadap 4 artikel jurnal lengkap (format Pdf) dengan ukuran file yang berbeda, yang dipilih secara acak dari situs penyedia jurnal Universitas Udayana. Hasil rata-rata dari pengukuran (data lengkap di Lampiran A) dapat dilihat pada Tabel 4.3 berikut :

Tabel 4.3. Waktu Eksekusi dan Memori untuk Aplikasi Fungsi ekstraksi Pdf.

Parameter	***ukuran file 56 kb***	***ukuran file 58 kb***	***ukuran file 123 kb***	***ukuran file 620 kb***
waktu eksekusi skrip keseluruhan (detik)	2.382915973663330	2.181618952751160	4.090649962425230	21.618898010253900
waktu eksekusi buka / baca data halaman pdf (detik)	2.367378973960880	2.171432995796200	3.986845946311950	21.568051004409800
waktu eksekusi konversi data pdf sampai dapat referensi (detik)	0.014953970909119	0.009585976600647	0.102544999122620	0.045914983749390
memory eksekusi awal sampai akhir (Byte)	90608	78056	240385.6	635940.8
memory eksekusi baca data halaman pdf (Byte)	111880	99160.8	277211.2	1452833.6

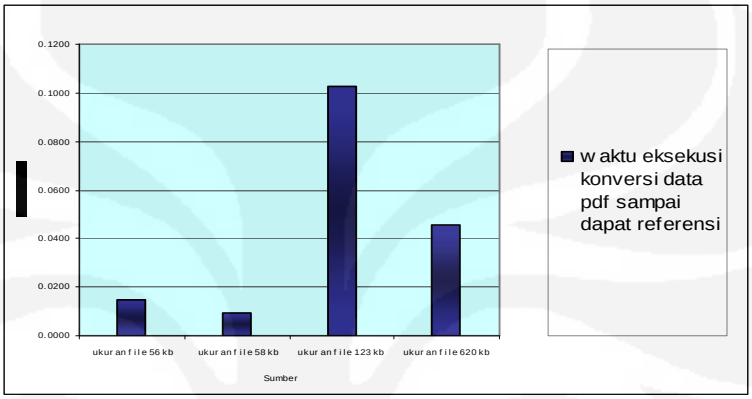
Dari data Tabel 4.3, dapat direpresentsikan dalam beberapa bentuk gambar grafik yang menggambarkan dengan jelas perbedaaan antara waktu dan memori dari masing-masing *file* yang dieksekusi pada saat aplikasi program dijalankan.



Gambar 4.9. Grafik Waktu Eksekusi Skrip Halaman Pdf Secara Keseluruhan Dan Saat Membuka atau Membaca File.

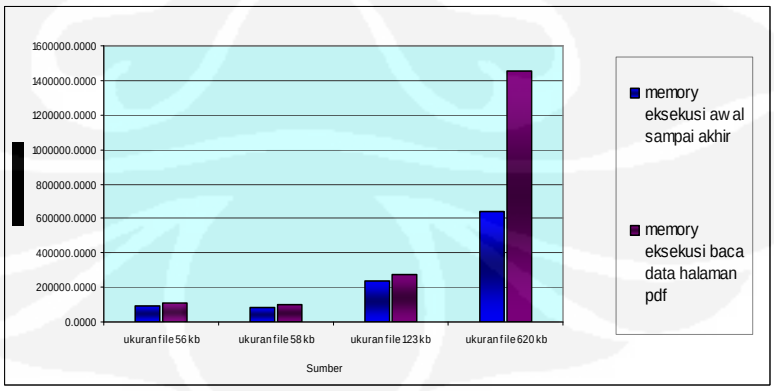
Dari grafik Gambar 4.9 di atas dapat diketahui terdapat perbedaan waktu eksekusi, dimana untuk waktu proses pembacaan halaman artikel lengkap pdf

tergantung dari besarnya ukuran file yang diproses. Sehingga idealnya semakin besar ukuran file akan semakin besar juga waktu yang diperlukan untuk eksekusi, dengan catatan bahwa kecepatan transfer data yang digunakan sama. Adanya waktu eksekusi yang lebih lama dengan ukuran file yang lebih kecil bisa disebabkan karena menurunnya kecepatan transfer data pada saat membuka file.



Gambar 4.10. Grafik Waktu Eksekusi Skrip untuk Mendapatkan Data Referensi

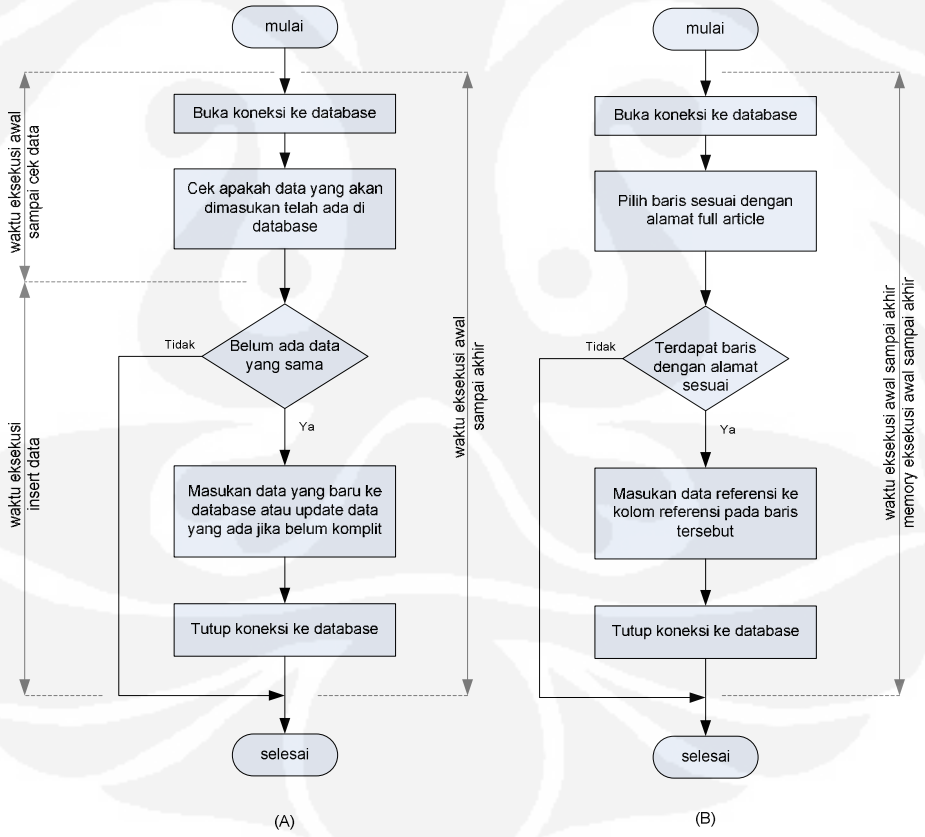
Dari grafik gambar 4.10 terlihat besar kecilnya waktu yang dibutuhkan untuk konversi data file pdf sampai didapatkan data referensi, tidak selalu ditentukan dari ukuran file, tetapi akan tergantung dari banyaknya ***stream*** pada file pdf yang diproses atau dikonversi sampai didapat data referensi atau daftar pustaka. Sehingga dengan semakin banyak stream yang ada pada file tersebut, akan mengakibatkan semakin lama waktu yang dibutuhkan .



Gambar 4.11. Grafik Penggunaan Memory pada Saat Skrip Ekstraksi Pdf dijalankan

Selain itu dalam penggunaan memori seperti terlihat pada grafik Gambar 4.11, diketahui dengan semakin besarnya ukuran file yang diproses akan mempengaruhi besarnya memori yang digunakan. Hal ini terkait dengan penggunaan variabel yang digunakan sebagai *buffer* untuk memproses data. Semakin besarnya file yang diproses akan bertambah besar juga memori yang diperlukan oleh aplikasi. Jika saat skrip dijalankan terdapat aplikasi lain pada komputer yang membutuhkan alokasi memori, sehingga akan menyebabkan penambahan penggunaan memori yang terhitung (menggunakan perintah `memory_get_usage()` pada php).

Informasi lain yang bisa didapatkan dari Aplikasi yang ada adalah mengenai informasi penggunaan waktu dan memori untuk memasukan data yang didapat ke database. Proses dari aplikasi ini dan titik pengukuran dapat terlihat pada diagram alir aplikasi fungsi input ke database pada Gambar 4.12.



Gambar 4.12 . Diagram alir Fungsi Input ke Database
(a) Input hasil Ekstraksi halaman web, (b) Input Hasil Ekstraksi Pdf

Hasil rata-rata pengukuran yang dilakukan 10 kali (data lengkap di Lampiran A) dari aplikasi sesuai diagram alir Gambar 4.12 dapat dilihat pada Tabel 4.4 dan Tabel 4.5. berikut:

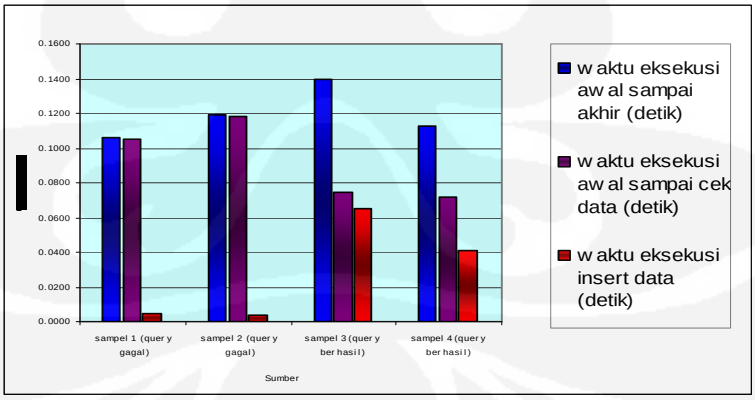
Tabel 4.4. Waktu Eksekusi Untuk Skrip Insert Data Hasil Ekstraksi Halaman Web Ke Database.

Parameter	***sampel 1 (query gagal)***	***sampel 2 (query gagal)***	***sampel 3 (query berhasil)***	***sampel 4 (query berhasil)***
waktu eksekusi awal sampai akhir (detik)	0.106240034103394	0.118740081787109	0.139830112457275	0.112760066986084
waktu eksekusi awal sampai cek data (detik)	0.105550050735474	0.118050098419189	0.074179887771606	0.071269989013672
waktu eksekusi insert data (detik)	0.004220008850098	0.003489971160889	0.065310001373291	0.041120052337647

Tabel 4.5. Waktu Eksekusi dan Memori untuk Skrip Insert Data Referensi ke Database.

Parameter	*647 karakter*	*1524 karakter*	*1990 karakter*	*2608 karakter*
waktu eksekusi awal sampai akhir (detik)	0.191042995	0.209623957	0.211986041	0.217761016
memory eksekusi awal sampai akhir (Byte)	775.2	1640	2104	2827.2

Untuk memperjelas Penggambaran waktu eksekusi insert data hasil ekstraksi halaman web ke database sesuai dengan Tabel 4.4 diperlihatkan pada Grafik 4.13. di bawah ini :

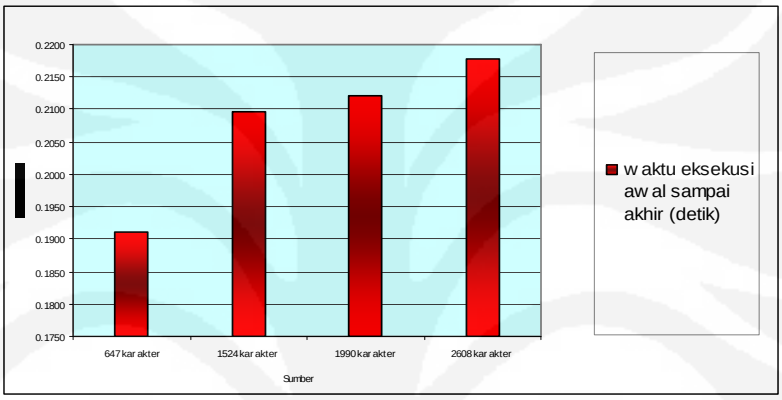


Gambar 4.13. Grafik Waktu Eksekusi Skrip insert data hasil ekstraksi web ke Database

Dari Gambar 4.13, diketahui terdapat perbedaan antara waktu eksekusi antara proses yang insert datanya berhasil dan tidak. Hal ini disebabkan karena

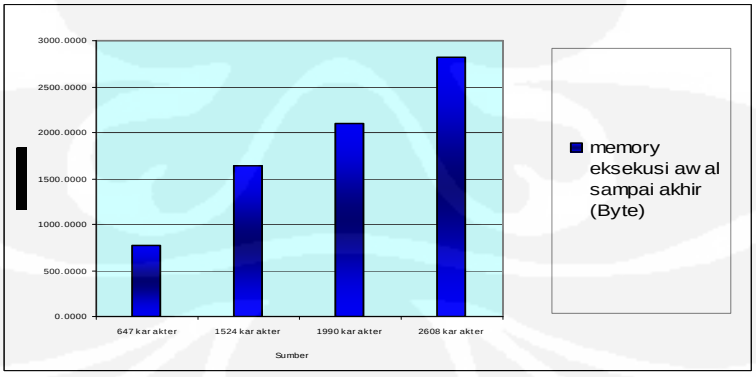
pada saat tidak berhasil dilakukan insert data, data yang sama telah ada sehingga pointer memilih baris pada data tersebut pada saat pengecekan data. Hal ini membutuhkan waktu yang lebih lama pada saat tersebut dan lebih sedikit pada saat proses insert (karena tidak melakukan insert data). Begitu juga sebaliknya untuk proses yang insert datanya berhasil.

Untuk menggambarkan waktu eksekusi proses insert data referensi ke database sesuai Tabel 4.5 dapat dilihat pada grafik Gambar 4.14.



Gambar 4.14. Grafik Waktu Eksekusi Skrip Insert Data Hasil Ekstraksi Pdf Ke Database

Untuk menggambarkan penggunaan memori saat eksekusi proses insert data referensi ke database sesuai Tabel 4.5 dapat dilihat pada grafik Gambar 4.15.

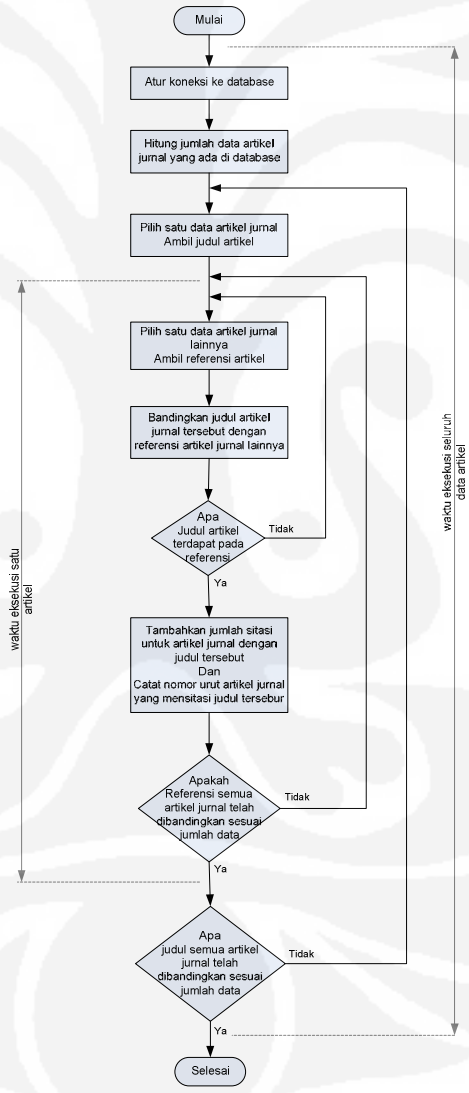


Gambar 4.15. Grafik Memori Eksekusi Skrip Insert Data Hasil Ekstraksi Pdf Ke Database

Dari grafik pada Gambar 4.14 dan 4.15 dapat diketahui bahwa dengan semakin banyak karakter atau semakin besarnya ukuran data yang dimasukan ke

database akan menggunakan waktu dan memori yang lebih besar. Memori besar ini karena variabel yang digunakan pada aplikasi program akan menyimpan data yang lebih besar.

Selain pengukuran yang telah dilakukan di atas, pengukuran juga dilakukan pada aplikasi fungsi pencarian sitasi jurnal yang berfungsi untuk mencari keterkaitan sitasi antara judul artikel jurnal yang terdapat dalam database. Proses dari aplikasi ini dan titik pengukuran dapat terlihat pada diagram alir Gambar 4.16.



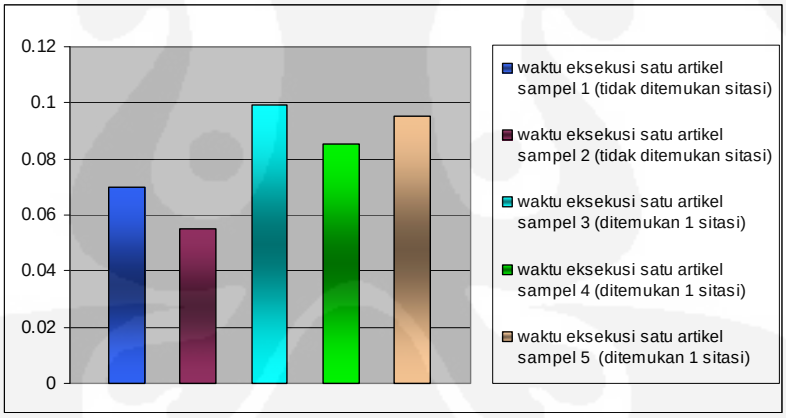
Gambar 4.16. Diagram Alir Fungsi Pencarian Sitasi Jurnal

Hasil eksekusi fungsi pencarian sitasi jurnal sesuai Gambar 4.16, yang dilakukan dengan jumlah data sebanyak 3358 data artikel jurnal. Didapatkan jumlah artikel yang disitasi sebanyak 39 artikel dengan jumlah sitasi terbanyak 4 sitasi untuk satu artikel jurnal. Hasil pengukuran yang dilakukan pada eksekusi fungsi di atas, didapatkan data seperti terlihat pada Tabel 4.6.

Tabel 4.6. Waktu Eksekusi Skrip untuk Mencari Data Sitasi.

Parameter	***Waktu (detik)***
waktu eksekusi seluruh data artikel	624.7453899
waktu eksekusi satu artikel sampel 1 (tidak ditemukan sitasi)	0.069890022
waktu eksekusi satu artikel sampel 2 (tidak ditemukan sitasi)	0.055279971
waktu eksekusi satu artikel sampel 3 (ditemukan 1 sitasi)	0.099380016
waktu eksekusi satu artikel sampel 4 (ditemukan 1 sitasi)	0.085209846
waktu eksekusi satu artikel sampel 5 (ditemukan 1 sitasi)	0.095350027

Untuk memperjelas Penggambaran waktu eksekusi proses pencarian sitasi jurnal sesuai Tabel 4.6 dapat dilihat pada grafik Gambar 4.14.



Gambar 4.17. Grafik Waktu Eksekusi Pencarian data Sitasi Antar Artikel Jurnal

Dari Gambar 4.17 di atas dapat diketahui bahwa waktu yang dibutuhkan untuk eksekusi atau proses satu data artikel jurnal tergantung dari urutan data tersebut pada database dan juga ditemukan tidaknya artikel lain yang mensitasi artikel tersebut (karena ada proses memasukan data artikel yang mensitasi jika terdapat data artikel yang mensitasi). Selain itu waktu yang terdapat pada Gambar

4.17 di atas akan bergantung pada banyaknya data yang diproses, dengan jumlah data sebanyak 3358 data artikel jurnal.

Semua data yang diperlukan dari artikel-artikel jurnal, didapatkan dari proses ektraksi pada halaman web dan artikel jurnal lengkap (*pdf file*). Data tersebut disimpan pada tabel yang terdapat pada database untuk selanjutnya dilakukan proses pencarian keterkaitan sitasi. Tipe data yang digunakan pada tabel di atas antara lain sebagai berikut :

- Kolom Nomor berisi nomor urut penyimpanan data artikel pada database, tipe data dari kolom ini adalah INT.
- Kolom Judul berisi nama dari judul-judul artikel yang didapat dari situs-situs penyedia jurnal, tipe data dari kolom ini adalah TEXT.
- Kolom Penulis berisi nama-nama penulis dari judul artikel bersangkutan, tipe data dari kolom ini adalah TEXT.
- Kolom Insitusi berisi nama institusi dari penulis artikel tersebut, tipe data dari kolom ini adalah TEXT.
- Kolom Penerbit berisi nama Penerbit dimana artikel pada jurnal tersebut dipublikasikan, tipe data dari kolom ini adalah TEXT.
- Kolom Abstrak berisi abstrak dari judul artikel terkait sebanyak ± 250 kata, tipe data dari kolom ini adalah TEXT.
- Kolom Alamat berisi alamat web dari artikel jurnal dengan judul tersebut, tipe data dari kolom ini adalah TEXT.
- Kolom Referensi berisi bagian referensi / daftar pustaka dari artikel jurnal dengan judul tersebut, tipe data dari kolom ini adalah LONG TEXT.
- Kolom Total Sitasi berisi jumlah sitasi yang telah dilakukan terhadap artikel jurnal dengan judul tersebut, tipe data dari kolom ini adalah INT.
- Kolom Disitasi berisi nomor-nomor urut artikel jurnal pada tabel database yang telah mensitasi jurnal judul tersebut, tipe data dari kolom ini adalah TEXT.

Ilustrasi tabel pada database yang digunakan untuk menyimpan data seluruh artikel jurnal yang didapat, dapat dilihat pada Tabel 4.7 berikut:

Tabel.4.7. Tabel Datajurnal

Nomer	Judul	Penulis	Institusi	Penerbit	Abstrak	Alamat	Referensi	Total Sitasi	Disitasi
1	Simulation of Active Filtering Applied to A Computer Centre	Julius Sentosa Setiadji1, Henry Hosiana Tumkeleka1 ,2	Electrical Engineering Department, Faculty of Industrial Technology, Petra Christian University.	Jurnal Teknik Electro Vol. 2, No 2, September 2002: 105 - 110	Dalam makalah ini dijelaskan tentang penggunaan "filter aktif type paralel dengan	http://puslit2.petra.ac.id/journal/index.php/civil/article/viewfile/17024/16992	[1] M. El-Habrouk, M.K. Darwish and P. Mehta, Active Power Filter: A Review, IEE	2	0;12
2	Kultur Campuran dan Faktor Lingkungan Mikroorganisme yang Berperan dalam Fermentasi "Tea-Cider"	Pingkan Aditiwati1 & Kusnadi2	1Departemen Biologi – FMIPA Institut Teknologi Bandung 2Jurusan Biologi – FMIPA Universitas Pendidikan Indonesia	PROC. ITB Sains & Tek. Vo. 35 A, No 2, 2003, 147-162	Teh merupakan hasil pertanian yang mengandung senyawa berkhasiat, terutama dalam bidang kesehatan.	http://proceedings.itb.ac.id/download.php?file=A0100_.pdf&id=503&up=10	1. Atlas, R. M., Parks, L. C. (ed), Handbook of Mikrobiologic al Media, CRC Press, Inc., Boca Raton Ann Arbor, London, Tokyo, pp.	1	44
3	PEMODELAN DAN SIMULASI KATALITIK KONVERTER PACKED BED UNTUK MENGOKSIDASI JELAGA PADA GAS BUANG KENDARAAN BERMESIN	M Nasikin1, Praswasti P.D.K. Wulan2 dan Yita Andriany	Program Studi Teknik Kimia, Departemen Teknik Gas dan Petrokimia, Fakultas Teknik, Universitas Indonesia, Depok 16424, Indonesia 1. Chemical	MAKARA, TEKNOLOGI, VOL. 8, NO. 3, DESEMBER 2004: 69-76	Kendaraan bermesin diesel banyak digunakan di Indonesia. Kendaraan jenis ini mengeluarkan polutan terutama jelaga	http://journal.ui.ac.id/?hal=download&q=397	[1] M. Nasikin, Pemanfaatan Catalytic Converter Untuk Kendaraan Bermotor di Indonesia, Laporan, Pusat Penelitian Sains dan	3	8;35.70;
4	- dst								

1.17 Pengujian Kinerja Halaman Antarmuka.

Pengujian kinerja sistem halaman antarmuka dilakukan juga dengan cara menjalankan aplikasi program. Aplikasi program yang dijalankan meliputi semua fungsi yang terdapat pada halaman antarmuka, baik halaman antamuka pembuka ataupun halaman antarmuka utama. Pengujian dilakukan meliputi beberapa fungsi berikut:

- Pengujian fungsi pencarian dengan kata kunci judul.
- Pengujian fungsi pencarian dengan kata kunci penulis.
- Pengujian fungsi melihat halaman selanjutnya.
- Pengujian fungsi melihat halaman sebelumnya.
- Pengujian fungsi melihat data artikel yang melakukan sitasi.
- Pengecekan hasil tampilan yang ada pada halaman antarmuka dan *link* untuk melihat artikel lengkap dari data artikel yang ditampilkan.

Hasil tampilan halaman antarmuka pengguna, yaitu halaman pembuka dapat dilihat pada Gambar 4.18 di bawah ini.

Gambar 4.19. Hasil Tampilan Halaman Antarmuka pembuka

Tampilan halaman antarmuka utama yang digunakan untuk menampilkan hasil pencarian data artikel jurnal, melakukan proses pencarian selanjutnya, dan melihat data artikel jurnal yang melakukan sitasi terhadap artikel jurnal lainnya, dapat dilihat pada Gambar 4.19 berikut ini:

Gambar 4.19. Hasil Tampilan Halaman Antarmuka utama

Dari tampilan halaman antarmuka dan dari percobaan yang dilakukan pada tiap fungsi-fungsi yang ada pada halaman antarmuka tersebut, diketahui semua fungsi dapat berjalan dengan baik

Selain pengujian fungsi-fungsi yang ada, dilakukan juga pengukuran kinerja halaman antarmuka, dimana cara yang dilakukan sama dengan pengukuran yang dilakukan pada sistem ekstraksi. Hasil pengukuran waktu eksekusi yang dilakukan pada beberapa aplikasi fungsi yang ada pada tampilan halaman utama dapat terlihat pada Tabel 4.8.

Tabel 4.8. Waktu Eksekusi Perintah-Perintah Pada Halaman utama.

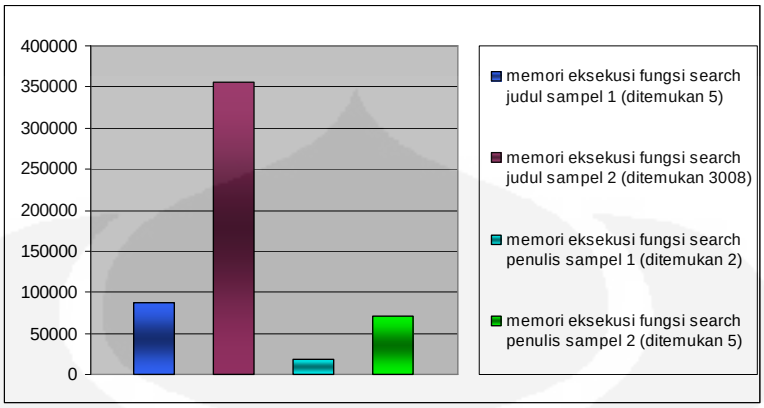
Parameter	***Waktu (detik)***
waktu eksekusi tampilan	0.008490085601807
waktu eksekusi fungsi search judul sampel 1 (ditemukan 5)	0.217839956283569
waktu eksekusi fungsi search judul sampel 2 (ditemukan 3008)	0.188859939575195
waktu eksekusi fungsi search penulis sampel 1 (ditemukan 2)	0.207910060882568
waktu eksekusi fungsi search penulis sampel 2 (ditemukan 5)	0.208349943161011
waktu eksekusi prev page	0.000050067901611
waktu eksekusi next page	0.000050067901611

Hasil pengukuran penggunaan memori saat proses eksekusi beberapa aplikasi fungsi yang ada pada tampilan halaman utama dapat terlihat pada Tabel 4.9.

Tabel 4.9. Memory Eksekusi Perintah-Perintah Pada Halaman utama

Parameter	***Ukuran (Byte)***
memori eksekusi fungsi search judul sampel 1 (ditemukan 5)	87064
memori eksekusi fungsi search judul sampel 2 (ditemukan 3008)	354952
memori eksekusi fungsi search penulis sampel 1 (ditemukan 2)	18160
memori eksekusi fungsi search penulis sampel 2 (ditemukan 5)	70224
memory eksekusi prev page sampel 1	88
memory eksekusi next page sampel 2	88
memory eksekusi tampilan	9904

Ilustrasi penggunaan memori pada salah satu fungsi yang ada pada tampilan antarmuka sesuai data dari tabel 4.9 dapat dilihat pada Gambar 4.18.



Gambar 4.20. Grafik Memori Eksekusi Pencarian Data Sesuai Kata Kunci

Dari grafik pada Gambar 4.20 terlihat, bahwa semakin banyak data yang ditampilkan atau ditemukan, maka akan semakin besar memori yang digunakan. Hal ini karena data jurnal yang ditemukan akan disimpan dalam variabel yang ada pada program selama sesi untuk halaman tersebut belum berakhir.

Dari proses implementasi dan pengujian diketahui bahwa penggunaan memori pada program harus diperhitungkan dengan baik. Hal ini diperlukan agar sistem tidak mengalami *memory exhaust* yang dapat mengakibatkan kesalahan dan berhentinya program aplikasi yang dijalankan. Salah satu pengaturan penggunaan memori ini salah satunya dengan *clear variabel* yang tidak terpakai setelah program berjalan atau menggunakan subroutine-subroutine pada program, kerena subroutine ini akan otomatis clear variabel lokal yang digunakan.

Pencarian yang dilakukan pada halaman antarmuka pengguna untuk menampilkan data yang sesuai dengan kata kunci yang dimasukan pengguna, dapat dikatakan sebagai *information retrieval system* yang merupakan suatu ilmu dalam pencarian suatu dokumen, pencarian informasi dalam dokumen, dan untuk metadata dari dokumen, baik dalam suatu relasional *database* ataupun dalam *World Wide Web*. Salah satu cara yang dapat digunakan untuk mengukur unjuk kerja *information retrieval system* adalah dengan mengevaluasi relevansi antara dokumen yang ditemukan yang sesuai dengan kebutuhan pengguna atau disebut *precision*.

$$precision = \frac{|\{relevant.documents\} \cap \{retrieved.documents\}|}{|\{retrieved.documents\}|} \quad \text{.....pers (1)}$$

Sehingga dari sistem yang diimplementasikan, dan dari pengujian dengan memasukan kata kunci berupa judul atau penulis didapatkan data sebagai berikut:

Tabel 4.10. Hasil Pengujian Aplikasi Pencarian Data

kata kunci	pilihan pencarian	*relevant document*	*retrieved document*
elemen	judul	18	18
Computer science	judul	11	11
jurnal	judul	4	4
ekstraksi	judul	4	4
teknologi	judul	51	51
yazid	penulis	2	2
joni	penulis	24	24
pranoto	penulis	1	1
agung	penulis	69	69
chandra	penulis	23	23
gunawan	penulis	29	29

Sesuai data yang didapat dan persamaan 1 di atas, maka dari hasil pencarian informasi yang dilakukan oleh sistem. Diketahui bahwa precision dari sistem adalah 1 atau seratus persen, dimana semua data yang berhasil di *retrieved* oleh sistem merupakan data yang relevan. Hal ini karena semua data yang berhasil di *retrieved* oleh sistem dan pada bagian judul atau penulis terdapat kata yang sesuai dengan kata kunci pencarian dianggap sebagai data yang relevan. Sehingga untuk sistem ini, perhitungan *precision* tidak dapat sepenuhnya diaplikasikan, karena sistem tidak memperhitungkan relevansi isi dokumen dengan kebutuhan pengguna, dan hanya memperhitungkan kesesuaian judul dan penulis dengan kata kunci yang dimasukan

1.18 Keterbatasan Sistem

Dari hasil output yang dihasilkan sistem dapat diketahui sistem belum dapat mengekstraksi semua tipe file pdf (yaitu pdf dibawah tipe 1.4). dan sumber situs penyedia jurnal elektronik yang digunakan terbatas 4 situs.

1.19 Pekerjaan Mendatang

Dari hasil pengujian sistem, diperlukan pengembangan lanjutan dari sistem yang telah ada. Pengembangan dilakukan dengan menelaah keterbatasan dan kekurangan yang ada pada sistem.

Agar tercapai sistem sitasi jurnal elektronik yang lengkap, diperlukan pengembangan dalam hal ekstraksi untuk semua tipe file pdf, diperlukan penambahan sumber-sumber institusi penerbit jurnal, dan aplikasi ekstraksi halaman web yang dapat mengikuti perubahan pada halaman web penyedia jurnal.

Selain itu diperlukan juga tambahan data-data yang ditampilkan pada halaman antarmuka, baik berupa perhitungan faktor dampak dan indeks lainnya.

BAB 5
KESIMPULAN

Setelah dilakukan perancangan, implementasi, uji coba dan analisa aplikasi Sistem Sitasi Jurnal Elektronik Indonesia yang dilakukan dengan mengambil informasi dari portal penyedia jurnal, dan membuat mashup gabungan informasi, dapat diambil beberapa kesimpulan :

- Setiap portal jurnal elektronik perguruan tinggi memiliki karakteristik yang berbeda-beda, sehingga perlu dibuat skrip (dengan PHP) ekstraksi tersendiri.
- Pengambilan data yang diperlukan dari tiap halaman web, dilakukan dengan cara mengenali tempat data disimpan dan *tag* yang digunakan.
- Hasil pengujian dan pengukuran sistem memperlihatkan bahwa :
  1. Semakin banyak jumlah artikel jurnal, dan juga jumlah kombinasi seri dan volume yang ada pada portal penyedia jurnal, akan mengakibatkan semakin besar waktu yang dibutuhkan untuk mengekstraksi seluruh data yang diperlukan dari situs tersebut.
  2. Semakin besar ukuran file yang diekstraksi pada aplikasi fungsi ekstraksi Pdf, maka akan semakin besar memori yang digunakan.
  3. Semakin besar ukuran data yang akan dimasukan dalam database, maka akan semakin besar waktu dan memori yang diperlukan untuk melakukan proses tersebut.
  4. Untuk mengekstraksi portal jurnal Institut Teknologi Bandung dibutuhkan waktu sekitar 332 detik dengan jumlah data yang berhasil diekstraksi sebanyak 183 artikel jurnal, portal jurnal Universitas Indonesia dibutuhkan waktu sekitar 504 detik dengan jumlah data yang berhasil diekstraksi sebanyak 370 artikel jurnal, portal jurnal Universitas Udayana dibutuhkan waktu sekitar 2314 detik dengan jumlah data yang berhasil diekstraksi sebanyak 1287 artikel jurnal, dan portal jurnal Universitas Kristen Petra dibutuhkan waktu sekitar

1731 detik dengan jumlah data yang berhasil diekstraksi sebanyak 1518 artikel jurnal.

- Secara umum sistem dapat bekerja dengan baik, dan bisa menampilkan keterkaitan sitasi antar artikel jurnal.
- Pengembangan sistem dapat dilakukan pada kemampuan untuk mengekstraksi halaman pdf dibawah tipe 1.4, dan penambahan jumlah portal penyedia jurnal *online* yang diekstraksi.

Daftar Acuan

[1]*Screen scraping*. http://en.wikipedia.org/wiki/ Screen_scraping. Diakses terakhir 3 Maret 2009.

[2]*Web scraping*. http://en.wikipedia.org/wiki/ Web_scraping. http://en.wikipedia.org /wiki/ Screen_scraping. Diakses terakhir 3 Maret 2009.

[3]Widyaseno, zulfikar, FX Ferdinand, Ruki Harwahyu dan Reza hadi S. Penggunaan Teknik Ekstraksi Web dalam Pengolahan referensi kepustakaan. Dept Teknik Elektro Universitas Indonesia

[4]S*emantic web*. http://en.wikipedia.org/wiki/Semantic_Web. Diakses terakhir 23 Februari 2009

[5]Palmer, sean B. *the semantic web: an introduction*. 2001. http://infomesh.net/2001/swintro/. Diakses terakhir 9 Februari 2009.

[6] *RDF Tutorial*. http://www.w3schools.com/rdf/. Diakses terakhir 10 februari 2009.

[7]*Resource Description Framework*. http://en.wikipedia.org/wiki/Resource _Description_Framework. Diakses terakhir 3 Februari 2009.

[8] X*ml tutorial*. http://www.tizag.com/xmlTutorial/. Diakses terakhir 6 februari 2009.

[9] Junaedi, Moh. *Pengantar XML*. 2003. http://ikc.vip.net.id/umum/junaedi-xml.php. Diakses terakhir 6 Februari 2009.

[10] *Mengenal Xml*. 2004. http://ictcenter-purwodadi.net/pustakamaya/files/disk1/10/ict-100-1001--guest-471-1-nopriant-l.pdf?PHPSESSID=58d0653df52f53dca122a68111f9ec7b. Diakses terakhir 5 Februari 2009.

[11] *XML*. http://en.wikipedia.org/wiki/XML. Diakses terakhir 6 februari 2009.

[12] *Kapow Mashup Server 6.3 Robomaker user guide*. Kapow technologies. 2007. http://www.kapowtech.com. Diakses terakhir 10 februari 2009.

[13] Kadir, abdul. *Dasar Pemrograman Web Dinamis Menggunakan PHP*. Yogyakarta. ANDI. 2008.

[14] *SQL*. http://id.wikipedia.org/wiki/SQL. Diakses terakhir 13 maret 2009

[15] *MySQL*. http://id.wikipedia.org/wiki/MySQL. Diakses terakhir 13 Maret 2009

[16]Amri, M Choirul. *Tutorial SQL (Structured Query Language).* 2003. http://ikc.vip.net.id/umum/choirul-sql.php 2003. Diakses terakhir 11 Maret 2009

[17] *Portable Document Format*. http://en.wikipedia.org/wiki/PDF. Diakses terakhir 01 Juni 2009

[18]Pressmann, Roger S. Software Engineering A Practitioners Approach (sixth edition). New York. McGraw-Hill.

[19]*Unified modeling language*. http://en.wikipedia.org/wiki/Unified_Modeling _Language. Diakses terakhir 14 Maret 2009

[20]*Mashup (web application hybrid).* http://en.wikipedia.org/wiki/Mashup_(web_ application_hybrid). Diakses terakhir 7 April 2009.

[21]*Citation*. http://en.wikipedia.org/wiki/Citation. Diakses terakhir 27 maret 2009

[22]*Citation Index*. http://en.wikipedia.org/wiki/Citation_index. Diakses terakhir 27 Maret 2009.

[23]*Panduan pencantuman sitasi bibliografis*. 2007. www.usu.ac.id /id/files/ panduan/pencantuman_sitasi.pdf. Diakses terakhir tanggal 27 Maret 2009.

[24]Iskandarsjah, kosasih. 2008. *Penghitungan? Impact Factor?*. http://www. edu2000.org/portal/index2.php?option=com_content&do_pdf=1&id=417. Diakses terakhir tanggal 27 Maret 2009.

[25] Scopus. www.scopus.com . Diakses terakhir tanggal 12 Juni 2009

[26] CiteSeerX. http://citeseerx.ist.psu.edu. Diakses terakhir tanggal 12 Juni 2009

[27]Google Scholar. http://scholar.google.co.id/intl/en/scholar/about.html. Diakses terakhir tanggal 12 Juni 2009.

[28]Lixto Visual Developer, http://www.lixto.com/images/Flash/vd_en/ vd_en .html. Diakses terakhir tanggal 14 Juni 2009.

[29]*Information Retrieval*, http://en.wikipedia.org/wiki/Information_retrieval.. Diakses terakhir 26 Juni 2009.

LAMPIRAN 1. Tabel Hasil Pengukuran Eksekusi Skrip

Tabel Hasil Pengukuran Waktu Eksekusi dan Memori untuk Skrip Insert Data Referensi ke Database

Sampling ke 1

Parameter	*647 karakter*	*1524 karakter*	*1990 karakter*	*2608 karakter*
waktu eksekusi awal sampai akhir (detik)	0.203350067	0.170929909	0.172139883	0.185009956
memory eksekusi awal sampai akhir (Byte)	768	1640	2104	3696

Sampling ke 2

Parameter	*647 karakter*	*1524 karakter*	*1990 karakter*	*2608 karakter*
waktu eksekusi awal sampai akhir (detik)	0.203579903	0.182509899	0.201730013	0.182160139
memory eksekusi awal sampai akhir (Byte)	768	1640	2104	2728

Sampling ke 3

Parameter	*647 karakter*	*1524 karakter*	*1990 karakter*	*2608 karakter*
waktu eksekusi awal sampai akhir (detik)	0.203169823	0.203449965	0.18039012	0.193100214
memory eksekusi awal sampai akhir (Byte)	768	1640	2104	2728

Sampling ke 4

Parameter	*647 karakter*	*1524 karakter*	*1990 karakter*	*2608 karakter*
waktu eksekusi awal sampai akhir (detik)	0.157520056	0.226059914	0.281270027	0.293040037
memory eksekusi awal sampai akhir (Byte)	768	1640	2104	2728

Sampling ke 5

Parameter	*647 karakter*	*1524 karakter*	*1990 karakter*	*2608 karakter*
waktu eksekusi awal sampai akhir (detik)	0.183860064	0.248549938	0.159790039	0.18073988
memory eksekusi awal sampai akhir (Byte)	768	1640	2104	2728

Sampling ke 6

Parameter	*647 karakter*	*1524 karakter*	*1990 karakter*	*2608 karakter*
waktu eksekusi awal sampai akhir (detik)	0.226099968	0.268249989	0.238070011	0.157579899
memory eksekusi awal sampai akhir (Byte)	768	1640	2104	2728

Sampling ke 7

Parameter	*647 karakter*	*1524 karakter*	*1990 karakter*	*2608 karakter*
waktu eksekusi awal sampai akhir (detik)	0.236760139	0.157999992	0.205659866	0.258409977
memory eksekusi awal sampai akhir (Byte)	840	1640	2104	2728

Sampling ke 8

Parameter	*647 karakter*	*1524 karakter*	*1990 karakter*	*2608 karakter*
waktu eksekusi awal sampai akhir (detik)	0.165549994	0.270910025	0.315020084	0.27038002
memory eksekusi awal sampai akhir (Byte)	768	1640	2104	2752

Sampling ke 9

Parameter	*647 karakter*	*1524 karakter*	*1990 karakter*	*2608 karakter*
waktu eksekusi awal sampai akhir (detik)	0.171569824	0.18866992	0.204820156	0.235549927
memory eksekusi awal sampai akhir (Byte)	768	1640	2104	2728

Sampling ke 10

Parameter	*647 karakter*	*1524 karakter*	*1990 karakter*	*2608 karakter*
waktu eksekusi awal sampai akhir (detik)	0.158970118	0.178910017	0.160970211	0.22164011
memory eksekusi awal sampai akhir (Byte)	768	1640	2104	2728

Tabel Hasil Pengukuran Waktu Eksekusi dan Memori untuk Aplikasi Fungsi ekstraksi Pdf

Sampling ke 1

Parameter	***ukuran file 56 kb***	***ukuran file 58 kb***	***ukuran file 123 kb***	***ukuran file 620 kb***
waktu eksekusi skrip keseluruhan (detik)	2.565870047	1.931349993	3.825269938	21.41845012
waktu eksekusi buka / baca data halaman pdf (detik)	2.549809933	1.919700146	3.719449997	21.3756001
waktu eksekusi konversi data pdf sampai dapat referensi (detik)	0.015429974	0.01101017	0.104529858	0.038399935
memory eksekusi awal sampai akhir (Byte)	125216	78064	255592	635800
memory eksekusi baca data halaman pdf (Byte)	145072	107448	288984	1452792

Sampling ke 2

Parameter	***ukuran file 56 kb***	***ukuran file 58 kb***	***ukuran file 123 kb***	***ukuran file 620 kb***
waktu eksekusi skrip keseluruhan (detik)	1.774909973	2.065939903	4.547060013	21.32848001
waktu eksekusi buka / baca data halaman pdf (detik)	1.758929968	2.055999994	4.441220045	21.28759003
waktu eksekusi konversi data pdf sampai dapat referensi (detik)	0.015409946	0.009340048	0.104599953	0.036949873
memory eksekusi awal sampai akhir (Byte)	81920	78008	236304	635968
memory eksekusi baca data halaman pdf (Byte)	103616	106936	273912	1452848

Sampling ke 3

Parameter	***ukuran file 56 kb***	***ukuran file 58 kb***	***ukuran file 123 kb***	***ukuran file 620 kb***
waktu eksekusi skrip keseluruhan (detik)	1.709189892	2.055459976	4.092730045	21.64299989
waktu eksekusi buka / baca data halaman pdf (detik)	1.693949938	2.04535985	3.990149975	21.60384989
waktu eksekusi konversi data pdf sampai dapat referensi (detik)	0.01466012	0.009500027	0.101320028	0.035219908
memory eksekusi awal sampai akhir (Byte)	81936	77944	236664	635976
memory eksekusi baca data halaman pdf (Byte)	103760	106904	274504	1452816

Sampling ke 4

Parameter	***ukuran file 56 kb***	***ukuran file 58 kb***	***ukuran file 123 kb***	***ukuran file 620 kb***
waktu eksekusi skrip keseluruhan (detik)	2.224309921	2.007049799	3.905779839	21.41992998
waktu eksekusi buka / baca data halaman pdf (detik)	2.20875001	1.997259855	3.801859856	21.37945008
waktu eksekusi konversi data pdf sampai dapat referensi (detik)	0.014969826	0.009200096	0.102670193	0.036469936
memory eksekusi awal sampai akhir (Byte)	81912	78144	236664	635984
memory eksekusi baca data halaman pdf (Byte)	103448	106728	274360	1452776

Sampling ke 5

Parameter	***ukuran file 56 kb***	***ukuran file 58 kb***	***ukuran file 123 kb***	***ukuran file 620 kb***
waktu eksekusi skrip keseluruhan (detik)	1.794610023	2.361320019	3.897089958	21.62764001
waktu eksekusi buka / baca data halaman pdf (detik)	1.779320002	2.351320028	3.794359922	21.58688998
waktu eksekusi konversi data pdf sampai dapat referensi (detik)	0.014719963	0.009399891	0.101449966	0.036750078
memory eksekusi awal sampai akhir (Byte)	82056	78120	236704	635976
memory eksekusi baca data halaman pdf (Byte)	103504	28728	274296	1452936

Sampling ke 6

Parameter	***ukuran file 56 kb***	***ukuran file 58 kb***	***ukuran file 123 kb***	***ukuran file 620 kb***
waktu eksekusi skrip keseluruhan (detik)	3.898169994	2.499429941	4.360839844	22.18309021
waktu eksekusi buka / baca data halaman pdf (detik)	3.882790089	2.489039898	4.257210016	22.10741019
waktu eksekusi konversi data pdf sampai dapat referensi (detik)	0.01481986	0.009789944	0.102400064	0.068949938
memory eksekusi awal sampai akhir (Byte)	125216	78064	255592	635800
memory eksekusi baca data halaman pdf (Byte)	145072	107448	288984	1452792

Sampling ke 7

Parameter	***ukuran file 56 kb***	***ukuran file 58 kb***	***ukuran file 123 kb***	***ukuran file 620 kb***
waktu eksekusi skrip keseluruhan (detik)	2.688899994	2.366780043	4.261420012	21.46690989
waktu eksekusi buka / baca data halaman pdf (detik)	2.673669815	2.356760025	4.157649994	21.42723989
waktu eksekusi konversi data pdf sampai dapat referensi (detik)	0.01466012	0.009439945	0.102540016	0.036100149
memory eksekusi awal sampai akhir (Byte)	81920	78008	236304	635968
memory eksekusi baca data halaman pdf (Byte)	103616	106936	273912	1452848

Sampling ke 8

Parameter	***ukuran file 56 kb***	***ukuran file 58 kb***	***ukuran file 123 kb***	***ukuran file 620 kb***
waktu eksekusi skrip keseluruhan (detik)	2.335530043	2.125669956	3.97874999	21.88532996
waktu eksekusi buka / baca data halaman pdf (detik)	2.319250107	2.115740061	3.875929832	21.8447299
waktu eksekusi konversi data pdf sampai dapat referensi (detik)	0.015670061	0.009339809	0.101549864	0.035949945
memory eksekusi awal sampai akhir (Byte)	81936	77944	236664	635976
memory eksekusi baca data halaman pdf (Byte)	103760	106904	274504	1452816

Sampling ke 9

Parameter	***ukuran file 56 kb***	***ukuran file 58 kb***	***ukuran file 123 kb***	***ukuran file 620 kb***
waktu eksekusi skrip keseluruhan (detik)	2.356339931	2.410539865	3.913939953	21.58199
waktu eksekusi buka / baca data halaman pdf (detik)	2.341229916	2.40038991	3.81069994	21.50801992
waktu eksekusi konversi data pdf sampai dapat referensi (detik)	0.014539957	0.009539843	0.101969957	0.066989899
memory eksekusi awal sampai akhir (Byte)	81912	78144	236664	635984
memory eksekusi baca data halaman pdf (Byte)	103448	106728	274360	1452776

Sampling ke 10

Parameter	***ukuran file 56 kb***	***ukuran file 58 kb***	***ukuran file 123 kb***	***ukuran file 620 kb***
waktu eksekusi skrip keseluruhan (detik)	2.481329918	1.992650032	4.123620033	21.63416004
waktu eksekusi buka / baca data halaman pdf (detik)	2.466089964	1.982760191	4.019929886	21.55973005
waktu eksekusi konversi data pdf sampai dapat referensi (detik)	0.014659882	0.009299994	0.102420092	0.067370176
memory eksekusi awal sampai akhir (Byte)	82056	78120	236704	635976
memory eksekusi baca data halaman pdf (Byte)	103504	106848	274296	1452936

Tabel Hasil Pengukuran Waktu Eksekusi untuk Aplikasi Fungsi Ekstraksi Halaman Web

Sampling ke 1

Parameter	***PETRA (detik)***	***UI (detik)***	***UDAYANA (detik)***	***ITB (detik)***
waktu eksekusi seluruh skrip crawler	1637.84693	491.38643	2235.96292	319.9883001
waktu eksekusi seluruh seri	1627.93839	489.83476	2233.97271	319.00564
waktu eksekusi seluruh volume	1627.06201	489.120015	2233.11734	317.04511
waktu eksekusi dari awal sampai dapat alamat seri	5.78647995	3.103320122	1.990190029	0.982640028
waktu eksekusi dari awal sampai dapat alamat volume	6.662840128	4.532570124	2.845540047	2.943120003
waktu eksekusi dari awal sampai dapat abstraksi	7.136490107	7.288789988	3.784119844	3.820500135
waktu eksekusi hal abstraksi sampel 1	0.000790119	0.000779867	0.000419855	0.002870083
waktu eksekusi hal abstraksi sampel 2	0.000749826	0.000789881	0.000400066	0.004469872
waktu eksekusi hal abstraksi sampel 3	0.000750065	0.000790119	0.000380039	0.002889872
waktu eksekusi hal abstraksi sampel 4	0.000789881	0.000790119	0.000360012	0.002810001
waktu eksekusi hal abstraksi sampel 5	0.000760078	0.000789881	0.000360012	0.004870176

Sampling ke 2

Parameter	***PETRA (detik)***	***UI (detik)***	***UDAYANA (detik)***	***ITB (detik)***
waktu eksekusi seluruh skrip crawler	1796.44753	530.187925	2418.7822	327.0608499
waktu eksekusi seluruh seri	1791.66211	528.639485	2416.78194	326.0107698
waktu eksekusi seluruh volume	1785.84316	527.939715	2415.88976	324.08044
waktu eksekusi dari awal sampai dapat alamat seri	4.785379887	3.096859932	2.000250101	1.050070047
waktu eksekusi dari awal sampai dapat alamat volume	6.568839788	4.496379852	2.89241004	2.980350018
waktu eksekusi dari awal sampai dapat abstraksi	7.701629877	7.323189974	3.848610163	3.980489969
waktu eksekusi hal abstraksi sampel 1	0.004309893	0.000770092	0.000420094	0.002860069
waktu eksekusi hal abstraksi sampel 2	0.016330004	0.00075984	0.000400066	0.004490137
waktu eksekusi hal abstraksi sampel 3	0.002909899	0.001569986	0.000380039	0.002900124
waktu eksekusi hal abstraksi sampel 4	0.001940012	0.000800133	0.000370026	0.002719879
waktu eksekusi hal abstraksi sampel 5	0.005210161	0.000789881	0.000369787	0.004950047

Sampling ke 3

Parameter	*PETRA (detik)*	*UI (detik)*	*UDAYANA (detik)*	*ITB (detik)*
waktu eksekusi seluruh skrip crawler	1664.9223	494.2032449	2222.07464	339.7818
waktu eksekusi seluruh seri	1660.92489	492.378135	2220.20189	338.21613
waktu eksekusi seluruh volume	1655.46404	491.6558551	2219.41843	335.7787001
waktu eksekusi dari awal sampai dapat alamat seri	3.997400045	3.65019989	1.87274003	1.56566
waktu eksekusi dari awal sampai dapat alamat volume	5.173609972	5.0947299	2.656179905	4.003040075
waktu eksekusi dari awal sampai dapat abstraksi	6.105560064	7.878340006	3.532989979	5.444669962
waktu eksekusi hal abstraksi sampel 1	0.002629995	0.000780106	0.000420094	0.002249956
waktu eksekusi hal abstraksi sampel 2	0.005340099	0.000499964	0.00026989	0.002769947
waktu eksekusi hal abstraksi sampel 3	0.004079819	0.000499964	0.000390053	0.001760006
waktu eksekusi hal abstraksi sampel 4	0.00248003	0.000499964	0.000329971	0.001650095
waktu eksekusi hal abstraksi sampel 5	1664.9223	494.2032449	2222.07464	339.7818

Sampling ke 4

Parameter	*PETRA (detik)*	*UI (detik)*	*UDAYANA (detik)*	*ITB (detik)*
waktu eksekusi seluruh skrip crawler	1765.9563	504.1042469	2321.08564	340.7718
waktu eksekusi seluruh seri	1761.93489	502.375132	2319.20188	339.23513
waktu eksekusi seluruh volume	1755.46404	501.3548511	2318.413431	333.9957001
waktu eksekusi dari awal sampai dapat alamat seri	4.995400045	3.65019989	2.17274003	1.05657501
waktu eksekusi dari awal sampai dapat alamat volume	5.283609672	4.9947289	2.996179905	4.003040075
waktu eksekusi dari awal sampai dapat abstraksi	6.515460064	7.575340006	3.943298598	5.444669962
waktu eksekusi hal abstraksi sampel 1	0.00386257	0.000760106	0.000390094	0.002347946
waktu eksekusi hal abstraksi sampel 2	0.007320077	0.000488964	0.00023988	0.002867957
waktu eksekusi hal abstraksi sampel 3	0.003079619	0.000469664	0.000370043	0.001860106
waktu eksekusi hal abstraksi sampel 4	0.00315003	0.000477838	0.000307571	0.001730197
waktu eksekusi hal abstraksi sampel 5	0.002910086	0.000501132	0.000351025	0.003022967

Sampling ke 5

Parameter	*PETRA (detik)*	*UI (detik)*	*UDAYANA (detik)*	*ITB (detik)*
waktu eksekusi seluruh skrip crawler	1717.14723	510.7871775	2327.37256	323.524575
waktu eksekusi seluruh seri	1709.80025	509.2371225	2325.377325	322.5082049
waktu eksekusi seluruh volume	1706.452585	508.529865	2324.50355	320.562775
waktu eksekusi dari awal sampai dapat alamat seri	5.285929918	3.100090027	1.995220065	1.016355038
waktu eksekusi dari awal sampai dapat alamat volume	6.615839958	4.514474988	2.868975043	2.96173501
waktu eksekusi dari awal sampai dapat abstraksi	7.419059992	7.305989981	3.816365004	3.900495052
waktu eksekusi hal abstraksi sampel 1	0.002550006	0.00077498	0.000419974	0.002865076
waktu eksekusi hal abstraksi sampel 2	0.008539915	0.00077486	0.000400066	0.004480004
waktu eksekusi hal abstraksi sampel 3	0.001829982	0.001180053	0.000380039	0.002894998
waktu eksekusi hal abstraksi sampel 4	0.001364946	0.000795126	0.000365019	0.00276494
waktu eksekusi hal abstraksi sampel 5	0.00298512	0.000789881	0.0003649	0.004910111

Sampling ke 6

Parameter	*PETRA (detik)*	*UI (detik)*	*UDAYANA (detik)*	*ITB (detik)*
waktu eksekusi seluruh skrip crawler	1730.684915	512.195585	2320.42842	333.421325
waktu eksekusi seluruh seri	1726.2935	510.50881	2318.491915	332.1134499
waktu eksekusi seluruh volume	1720.6536	509.797785	2317.654095	329.9295701
waktu eksekusi dari awal sampai dapat alamat seri	4.391389966	3.373529911	1.936495066	1.307865024
waktu eksekusi dari awal sampai dapat alamat volume	5.87122488	4.795554876	2.774294972	3.491695046
waktu eksekusi dari awal sampai dapat abstraksi	6.903594971	7.60076499	3.690800071	4.712579966
waktu eksekusi hal abstraksi sampel 1	0.003469944	0.000775099	0.000420094	0.002555013
waktu eksekusi hal abstraksi sampel 2	0.010835052	0.000629902	0.000334978	0.003630042
waktu eksekusi hal abstraksi sampel 3	0.003494859	0.001034975	0.000385046	0.002330065
waktu eksekusi hal abstraksi sampel 4	0.002210021	0.000650048	0.000349998	0.002184987
waktu eksekusi hal abstraksi sampel 5	0.004210114	0.000645041	0.000369906	0.003955007

Sampling ke 7

Parameter	*PETRA (detik)*	*UI (detik)*	*UDAYANA (detik)*	*ITB (detik)*
waktu eksekusi seluruh skrip crawler	1781.201915	499.1537459	2369.93392	340.2768
waktu eksekusi seluruh seri	1776.7985	497.3766335	2367.99191	338.72563
waktu eksekusi seluruh volume	1770.6536	496.5053531	2367.151596	334.8872001
waktu eksekusi dari awal sampai dapat alamat seri	4.890389966	3.65019989	2.086495066	1.311117505
waktu eksekusi dari awal sampai dapat alamat volume	5.92622473	5.0447294	2.944294972	4.003040075
waktu eksekusi dari awal sampai dapat abstraksi	7.108544971	7.726840006	3.89595438	5.444669962
waktu eksekusi hal abstraksi sampel 1	0.004086231	0.000770106	0.000405094	0.002298951
waktu eksekusi hal abstraksi sampel 2	0.011825041	0.000494464	0.000319973	0.002818952
waktu eksekusi hal abstraksi sampel 3	0.002994759	0.000484814	0.000375041	0.001810056
waktu eksekusi hal abstraksi sampel 4	0.002545021	0.000488901	0.000338798	0.001690146
waktu eksekusi hal abstraksi sampel 5	0.004060123	0.000500667	0.000360406	0.002991467

Sampling ke 8

Parameter	*PETRA (detik)*	*UI (detik)*	*UDAYANA (detik)*	*ITB (detik)*
waktu eksekusi seluruh skrip crawler	1741.551765	497.7453385	2324.2291	330.3800501
waktu eksekusi seluruh seri	1735.86757	496.104946	2322.289602	329.120385
waktu eksekusi seluruh volume	1730.958313	495.237433	2321.458491	325.5204051
waktu eksekusi dari awal sampai dapat alamat seri	5.140664982	3.376760006	2.083980048	1.019607519
waktu eksekusi dari awal sampai dapat alamat volume	5.949724815	4.763649512	2.932577474	3.473080039
waktu eksekusi dari awal sampai dapat abstraksi	6.967260028	7.432064997	3.879831801	4.632585049
waktu eksekusi hal abstraksi sampel 1	0.003206288	0.000769986	0.000405034	0.002609014
waktu eksekusi hal abstraksi sampel 2	0.007929996	0.000639422	0.000319973	0.003668914
waktu eksekusi hal abstraksi sampel 3	0.0024548	0.000629891	0.000375041	0.002374989
waktu eksekusi hal abstraksi sampel 4	0.002257488	0.000633978	0.000336295	0.002270099
waktu eksekusi hal abstraksi sampel 5	0.002947603	0.000645506	0.000357962	0.003946571

Sampling ke 9

Parameter	*PETRA (detik)*	*UI (detik)*	*UDAYANA (detik)*	*ITB (detik)*
waktu eksekusi seluruh skrip crawler	1723.916072	511.4913812	2323.90049	328.47295
waktu eksekusi seluruh seri	1718.046875	509.8729662	2321.93462	327.3108274
waktu eksekusi seluruh volume	1713.553093	509.163825	2321.078822	325.2461725
waktu eksekusi dari awal sampai dapat alamat seri	4.838659942	3.236809969	1.965857565	1.162110031
waktu eksekusi dari awal sampai dapat alamat volume	6.243532419	4.655014932	2.821635008	3.226715028
waktu eksekusi dari awal sampai dapat abstraksi	7.161327481	7.453377485	3.753582537	4.306537509
waktu eksekusi hal abstraksi sampel 1	0.003009975	0.000775039	0.000420034	0.002710044
waktu eksekusi hal abstraksi sampel 2	0.009687483	0.000702381	0.000367522	0.004055023
waktu eksekusi hal abstraksi sampel 3	0.00266242	0.001107514	0.000382543	0.002612531
waktu eksekusi hal abstraksi sampel 4	0.001787484	0.000722587	0.000357509	0.002474964
waktu eksekusi hal abstraksi sampel 5	0.003597617	0.000717461	0.000367403	0.004432559

Sampling ke 10

Parameter	*PETRA (detik)*	*UI (detik)*	*UDAYANA (detik)*	*ITB (detik)*
waktu eksekusi seluruh skrip crawler	1755.943415	494.5658842	2280.09601	336.8490625
waktu eksekusi seluruh seri	1751.546	492.969853	2278.131156	335.41954
waktu eksekusi seluruh volume	1745.6536	492.178724	2277.287915	332.4083851
waktu eksekusi dari awal sampai dapat alamat seri	4.640889966	3.240040064	2.037085038	1.309491264
waktu eksekusi dari awal sampai dapat alamat volume	5.898724805	4.648109818	2.88905876	3.747367561
waktu eksekusi dari awal sampai dapat abstraksi	7.006069971	7.360427492	3.831975823	5.078624964
waktu eksekusi hal abstraksi sampel 1	0.003778088	0.000774927	0.000412445	0.002426982
waktu eksekusi hal abstraksi sampel 2	0.011330046	0.000714652	0.00036002	0.003224497
waktu eksekusi hal abstraksi sampel 3	0.003244809	0.000710005	0.00037754	0.00207006
waktu eksekusi hal abstraksi sampel 4	0.002377521	0.000712049	0.000348154	0.001937567
waktu eksekusi hal abstraksi sampel 5	0.004135119	0.000717694	0.000358987	0.003473237

LAMPIRAN 2. Skrip PHP Aplikasi Program ekstraksi Pdf

```
echo ('awalpdf  '); echo microtime(); echo("<BR>");
echo ('mem_awalpdf  '); echo memory_get_usage(); echo("<BR>");
set_time_limit(6000);
ini_set('memory_limit', '300M');
$isi = file($alamatfull);
$isiokmain='';
foreach($isi as $baris)
       $isiokmain = $isiokmain . ($baris);
echo ('akhirambildata  '); echo microtime(); echo("<BR>");
echo ('mem_akhirambildata  '); echo memory_get_usage(); echo("<BR>");
$isiokmain=$isiokmain.'x';
$jum_endstream=substr_count($isiokmain,'endstream');
$type=substr($isiokmain,1,10);
echo ("$type<BR>");
$isiokmaincon=strrev($isiokmain);
 unset($isiokmain);
 unset($isi);
echo ('awalconvert  '); echo microtime(); echo("<BR>");
echo ('mem_awalconvert  '); echo memory_get_usage(); echo("<BR>");
for($jum_abc=1; $jum_abc <= $jum_endstream ; $jum_abc++):
   $posisiawalstream=strpos($isiokmaincon,strrev('endstream'));
   $posisiakhirstream=strpos(substr($isiokmaincon,$posisiawalstream+9,-
1),strrev('stream'));
   $isistream=trim(substr($isiokmaincon,$posisiawalstream+9,$posisiakhirstream));
   $isistreamok=strrev($isistream);
   $data=substr($isistreamok,0,2);
   $uncompressed='';
       if (((($data == chr(72) . chr(137))) Or (($data == chr(120) . chr(156))) Or
(substr($data,0,1)==chr(120))):
              // echo ("$isistreamok <BR><BR>");
              $uncompressed = @gzuncompress($isistreamok);
               // echo ("$uncompressed<BR><BR>");
       endif;
      unset($isistreamok);
       echo ('awalconverthasil  '); echo microtime(); echo("<BR>");
echo ('mem_awalconverthasil  '); echo memory_get_usage(); echo("<BR>");

//================================================================ konversi data
hasil pembacaan pdf file ( dalam kurung )
       $buffer=$uncompressed;
    unset($uncompressed);
       $jumlah=strlen($buffer);
       $bufferproses = $buffer . 'x' ;
       $lew1=chr(92) .'(' ;
       $lew2= chr(92) .')';
       $jumxx=substr_count($buffer,'(');
       $jumyy=substr_count($buffer,')');
if ( (abs($jumxx-$jumyy))<=100):
       $posisi1=1;
       $a=1;
       while ($a<$jumxx)
       {
              $posisi1 = strpos($bufferproses,'(');
              $posisi2 = strpos($bufferproses,')');
              $tes2=substr($bufferproses,($posisi2 - 1),2);
              if ($tes2 == $lew2 ):
                     $bufferprosessemx = $bufferproses;
                     $bufferprosesx=substr($bufferprosessemx,$posisi2+1,-1)."x";
                     $posisi2sem = strpos($bufferprosesx,')');
                     $posisi2 = $posisi2 + $posisi2sem + 1;
                     $tes2=substr($bufferproses,($posisi2 - 1),2);
                     if ($tes2 == $lew2 ):
                            $bufferprosessemx = $bufferproses;
                            $bufferprosesx=substr($bufferprosessemx,$posisi2+1,-
1)."x";
                            $posisi2sem = strpos($bufferprosesx,')');
                            $posisi2 = $posisi2 + $posisi2sem + 1;
                     endif;
              endif;
                     $posisi3 = $posisi2 + 1;
```

```
                    $panjang = $posisi2 - $posisi1 - 1;
                    $posisiawal = $posisi1 + 1;
                    $bufferisi = $bufferisi .
substr($bufferproses,$posisiawal,$panjang);
                    $bufferprosessem=$bufferproses;
                    $bufferproses=substr($bufferprosessem,$posisi3,-1)."x";
               $a=$a+1;
      $mem = memory_get_usage();
         echo ('xxxxx====  '); echo memory_get_usage();
      if ($mem >150000000):
            unset($bufferproses);
          unset($bufferisi);
          echo ('vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv====  ');
          break;
      endif;

       }

//echo ('akhirconverthasil  '); echo microtime(); echo("<BR>");
//echo ('mem_akhirconverthasil  '); echo memory_get_usage(); echo("<BR>");
       $a=0;
       $bufferisi=str_replace($lew1,'(',$bufferisi);
       $bufferisi=str_replace($lew2,')',$bufferisi);
       $bufferisi=strtolower($bufferisi);
else:
    $bufferisi='';
endif;

//=====================================================================

   if (strstr($bufferisi,chr(143))):
      $bufferisi='';
   endif;
       $bufferisiall = $bufferisi . $bufferisiall;
   if (strstr($bufferisiall,'acuan')):
      break;
   elseif (strstr($bufferisiall,'pustaka')):
      break;
   elseif (strstr($bufferisiall,'referen')):
     break;
   endif;
  $isiokmaincon=substr($isiokmaincon,$posisiawalstream+6+$posisiakhirstream+9,-1);
  unset($bufferisi);

endfor;
//echo("$bufferisiall");
 $posisiawalref=strpos($bufferisiall,'pustaka');
 if (!$posisiawalref)
    $posisiawalref=strpos($bufferisiall,'referen');
    if (!$posisiawalref)
       $posisiawalref=strpos($bufferisiall,'aftar acuan')-1;
 $cekposakhir= strpos(strrev($bufferisiall),strrev('j' .chr(10). 'et' . chr(10) .
'emc'));
 if (!$cekposakhir)
     $cekposakhir= strpos(strrev($bufferisiall),strrev('tj' .chr(13). 'et' .
chr(13) . 'j'));
 $posisiakhirref=strlen($bufferisiall)- $cekposakhir;
 $referensi=substr($bufferisiall,$posisiawalref,$posisiakhirref-$posisiawalref);
 if (strlen($referensi)<10)
 {
  if ((strlen($bufferisiall))>3100)
     $referensi=substr($bufferisiall,-3000,3000);
 }
 $referensi=str_replace(chr(13),'',$referensi);
 $referensi=str_replace(chr(10),'',$referensi);
 $referensi=str_replace('  ','',$referensi);
 for ($chargan=127; $chargan<=255 ; $chargan++):
     $referensi=str_replace(chr($chargan),'',$referensi);
 endfor;
 for ($chargan=1; $chargan<=31 ; $chargan++):

     $referensi=str_replace(chr($chargan),'',$referensi);
```

```
 endfor;

 for ($chargan=33; $chargan<=39 ; $chargan++):
     $referensi=str_replace(chr($chargan),'',$referensi);
 endfor;
for ($chargan=42; $chargan<=44 ; $chargan++):
    $referensi=str_replace(chr($chargan),'',$referensi);
 endfor;

echo ('akhirconvert  '); echo microtime(); echo("<BR>");
echo ('mem_akhirconvert  '); echo memory_get_usage(); echo("<BR>");
   // insertdataref($alamatfull, $referensi);
echo ('akhirinsert  '); echo microtime(); echo("<BR>");
echo ('mem_akhirinsert  '); echo memory_get_usage(); echo("<BR>");
echo ('akhirpdf '); echo microtime(); echo("<BR>");
echo ('mem_akhirpdf  '); echo memory_get_usage(); echo("<BR>");
    echo (" referensi : $referensi <BR><BR>");

}
```