

**Perancangan dan Implementasi *File Sharing Extension*  
pada *Browser Google Chrome***

**Artikel Ilmiah**



**Peneliti:**

**Agustinus Raharjo (672007212)**

**Hendry, M.Kom**

**Ramos Somya, S.Kom., M.Cs**

**Program Studi Teknik Informatika  
Fakultas Teknologi Informasi  
Universitas Kristen Satya Wacana  
Salatiga  
November 2012**


Perancangan dan Implementasi *File Sharing Extension* pada *Browser Google Chrome*


Oleh,  
Agustinus Raharjo  
NIM : 672007212

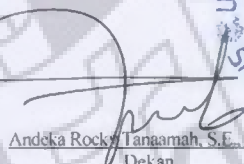
Artikel Ilmiah

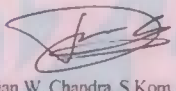
Diajukan Kepada Program Studi Teknik Informatika, Fakultas Teknologi Informasi guna memenuhi sebagian dari persyaratan untuk mencapai gelar Sarjana Komputer

Disetujui oleh.

  
Hendry, M.Kom.  
Pembimbing 1

  
Ramos Somya, S.Kom., M.Cs.  
Pembimbing 2

  
Andeka Rocky Tanaamah, S.E., M.Cs.  
Dekan

  
Dian W. Chandra, S.Kom., M.Cs.  
Ketua Program Studi

FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN SATYA WACANA  
SALATIGA  
1980



FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN SATYA WACANA  
Jalan Diponegoro 52 - 60  
Phone. (0298) 321212 (Hunting)  
Fax. (0298) 321433  
E-mail: fti@uksw.edu  
Salatiga 50711 - INDONESIA



### LEMBAR PERSETUJUAN PUBLISH JURNAL

Dengan mempertimbangkan isi dari jurnal mahasiswa :

Nama : Agustinus Raharjo

NIM : 672007212

Maka jurnal ini dinyatakan :

LAYAK TERBIT /  TIDAK LAYAK TERBIT

Menyetujui.

Pembimbing 1

(..... Hendry, M.Kom .....) )

Pembimbing 2

(..... Ramos Senya, S.Kom, M.Cs .....) )

Penguji 1

(..... Supitadi, S.Si, M.Kom .....) )

Penguji 2

(..... Frederik S. Pakilaya, S.Kom, M.Cs .....) )

1956



### PERNYATAAN TIDAK PLAGIAT DAN PERSETUJUAN AKSES

Sebagai sivitas akademik Universitas Kristen Satya Wacana, saya yang bertanda tangan di bawah ini:

Nama : AGUSTINUS RAHARJO  
NIM : 672007212 Email : dulharjo@yahoo.com  
Fakultas : TEKNOLOGI INFORMASI Program Studi : TEKNIK INFORMATIKA  
Judul tugas akhir : Perancangan dan Implementasi File sharing Extension pada Browser Google Chrome

Dengan ini menyerahkan karya tersebut di atas untuk disimpan dalam Koleksi Digital Perpustakaan Universitas dengan ketentuan akses tugas akhir elektronik sebagai berikut (beri tanda pada kotak yang sesuai):

- a. Saya mengizinkan karya tersebut diunggah ke dalam aplikasi Koleksi Digital Perpustakaan Universitas, dan/atau portal GARUDA.  
 b. Saya tidak mengizinkan karya tersebut diunggah ke dalam aplikasi Koleksi Digital Perpustakaan Universitas, dan/atau portal GARUDA. \*

\* poin b harus dilampiri dengan surat dari Dekan/ Kaprodi atau pembimbing TA dengan diketahui oleh pimpinan fakultas yang menjelaskan alasan pilihan. Yang akan ditampilkan adalah halaman judul + abstrak.

Dengan ini saya juga menyatakan bahwa:

1. Hasil karya yang saya serahkan ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar kesarjanaan baik di Universitas Kristen Satya Wacana maupun di institusi pendidikan lainnya.
2. Hasil karya saya ini bukan saduran/ terjemahan melainkan merupakan gagasan, rumusan, dan hasil pelaksanaan penelitian/ implementasi saya sendiri, tanpa bantuan pihak lain, kecuali arahan pembimbing akademik dan narasumber penelitian.
3. Hasil karya saya ini merupakan hasil revisi terakhir setelah diujikan yang telah diketahui dan disetujui oleh pembimbing.
4. Dalam karya saya ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali yang digunakan sebagai acuan dalam naskah dengan menyebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
5. Saya menyerahkan hak non-eksklusif kepada Perpustakaan Universitas – Universitas Kristen Satya Wacana untuk menyimpan, mengatur akses serta melakukan pengelolaan terhadap karya saya ini dengan mengacu pada ketentuan akses tugas akhir elektronik di atas dan norma hukum yang berlaku.

Pernyataan ini saya buat dengan sesungguhnya. Apabila di kemudian hari terbukti ada penyimpangan dan ketidakbenaran dalam pernyataan ini maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya saya ini, serta sanksi lain yang sesuai dengan ketentuan yang berlaku di Universitas Kristen Satya Wacana.

21 Februari 2013

Tanggal penyerahan

Hendry

Tanda tangan & nama terang pembimbing I

Mengetahui,

Agustinus Raharjo

Tanda tangan & nama terang mahasiswa

Ramos Sema

Tanda tangan & nama terang pembimbing II

# Perancangan dan Implementasi *File Sharing Extension* pada *Browser Google Chrome*

<sup>1)</sup> Agustinus Raharjo <sup>2)</sup> Hendry <sup>3)</sup> Ramos Somya

Program Studi Teknik Informatika  
Fakultas Teknologi Informasi  
Universitas Kristen Satya Wacana

Jl. Diponegoro 52-60, Salatiga 50711, Indonesia

E-mail: <sup>1)</sup> dulharjo@gmail.com, <sup>2)</sup> hendry.honk@gmail.com, <sup>3)</sup> ramos.somya@gmail.com

## Abstract

*Browser extension is an additional functionality that makes web browser work more optimally according to the user needs. The managing of the downloaded files in normal way is less effective, especially when the users want to share files with another users. To solve the problems, this research built an extension using Google Chrome platform. This tool is able to re-host objects such as images, links and documents on the web browser into the online storage. Online storage is managed on a web server that makes files stored more structured and can be re-used or shared to another user easily as long as connected to the internet.*

*Keywords : Google Chrome, Extension, Online Storage, File Sharing, Browser*

## Abstrak

*Browser extension merupakan fungsionalitas tambahan yang menjadikan web browser bekerja dengan lebih maksimal sesuai kebutuhan user. Pengelolaan file yang telah diunduh dengan cara biasa sangat kurang efektif, terlebih ketika user ingin berbagi file dengan user lain. Untuk mengatasi masalah seperti itu, pada penelitian ini dibangun sebuah tool berupa extension dengan platform Google Chrome. Tool ini dapat digunakan untuk melakukan re-host object pada web browser ke dalam online storage, object tersebut antara lain gambar, link dan file document. Online storage dikelola pada web server sehingga menjadikan file tersimpan dengan lebih terstruktur dan dapat digunakan kembali serta dibagikan kepada user lain dengan mudah kapan pun dan di mana pun selama terhubung dengan jaringan internet.*

*Kata Kunci : Google Chrome, Extension, Online Storage, File Sharing, Browser*

- 
- <sup>1)</sup> Mahasiswa Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana Salatiga.  
<sup>2)</sup> Staff Pengajar Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana Salatiga.  
<sup>3)</sup> Staff Pengajar Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana Salatiga.

## 1. Pendahuluan

Seiring meluasnya jaringan *internet* dan pemanfaatannya dalam berbagai bidang, *internet* secara langsung berdampak pada jumlah pengguna *web browser* yang semakin meningkat. Hal tersebut terjadi karena hampir seluruh aktifitas ber-*internet* dilakukan dengan menggunakan *web browser*. Dengan menggunakan *web browser*, pengguna *internet* dapat melakukan interaksi dengan data yang tersebar di *internet* dengan mudah.

Pemanfaatan *browser* dalam berselancar di dunia *internet* semakin dipermudah dengan adanya *add-on* atau *extension*. *Extension* merupakan alat tambahan yang dapat dipasang dalam sebuah *web browser*. Dengan menggunakan *extension*, selain pengguna dipermudah untuk menampilkan data yang diharapkan secara cepat, beban yang dijalankan oleh *browser* pun menjadi semakin berkurang.

*Google Chrome* merupakan *web browser* yang dikeluarkan oleh perusahaan *Google*. Dari pertama kali rilis di bulan Oktober 2008 sampai bulan Juni 2012 pengguna *Google Chrome* berkembang begitu pesat. Di awal rilisnya pengguna *Google Chrome* hanya sebesar 3.1% dari keseluruhan pengguna *browser*. Namun setelah berjalan beberapa waktu, *Google Chrome* menempati posisi pertama dengan 47,7% atau hampir separuh dari total pengguna *browser* [1].

*Google Chrome* juga mengizinkan pengguna untuk menambahkan atau mengembangkan *extension*-nya sendiri. Dengan jumlah pengguna *Google Chrome* yang begitu banyak, penggunaan *extension* untuk mempermudah dan mempercepat aktifitas *browsing* tentunya juga akan sangat diminati.

Perkembangan teknologi yang demikian pesat juga mempengaruhi pola penyimpanan suatu berkas, data-data hasil *browsing* sangat mungkin untuk digunakan tanpa harus menyimpannya terlebih dahulu ke dalam *local storage* melainkan dapat disimpan dalam suatu *cloud*. *Cloud* merupakan sistem penyimpanan data secara virtual yang memungkinkan pengguna dapat menggunakan data-data di dalam *cloud* kapanpun dan di mana pun selama terhubung dengan koneksi *internet*.

Dalam praktek sehari - hari kegiatan *browsing* yang sering dilakukan dalam penggunaan *internet* adalah *download* dan *upload file*, baik berupa gambar, *document* atau sekedar *bookmark link* suatu *website*. Oleh karena itu, melalui penelitian ini akan dibangun sebuah *tool* berupa *Google Chrome extension* yang digunakan untuk menyimpan *file* hasil *browsing* dengan cara *re-host* ke dalam suatu *online storage*. *File* yang disimpan di dalam *online storage* dapat dikelola dan digunakan dengan lebih baik serta dapat dibagikan kepada pengguna lain. Selain itu dengan membangun *web server* sendiri pengguna menjadi tahu di mana letak *file* mereka disimpan.

## 2. Tinjauan Pustaka

Pemanfaatan *extension* dalam memaksimalkan kinerja *Google Chrome browser* sangat bermanfaat bagi *internet surfer*. Dengan adanya *Google Chrome extension*, pengguna dapat menyesuaikan *browser* dengan fitur-fitur yang disukainya. *Music Downloader Extension* merupakan *Google Chrome extension* yang dibangun untuk mempermudah seorang pengguna *browser* dalam mencari dan mengunduh lagu dari *internet* berdasarkan penyanyi atau judul lagu. Dalam penelitian tersebut juga menyimpulkan bahwa dengan *Music Downloader Extension*, pencarian lirik lagu akan dapat dilakukan dengan mudah [2].

Berdasarkan pemanfaatan *Google Chrome Extension* tersebut maka diimplementasikan suatu *extension* yang dapat melakukan *re-host* berbagai *file* ke dalam *online storage* melalui suatu *link*. Dengan *extension* ini *user* akan lebih mudah dalam mengelola *file* yang di *download*, berbagi dengan *user* lain dan menggunakan kembali *file* dengan lebih baik.

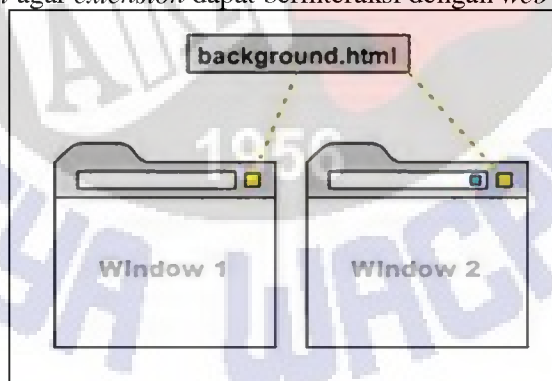
### Google Chrome Extension

Google Chrome extension merupakan browser extension yang memodifikasi Google Chrome browser. Browser extension ini ditulis dengan menggunakan teknologi web seperti HTML, JavaScript dan juga CSS [3]. Google Chrome extension menyimpan informasinya di dalam sebuah file manifest yang disebut dengan manifest.json. Informasi tersebut berupa source file apa saja yang akan difungsikan pada saat extension dijalankan. Contoh Manifest.json untuk mengambil informasi dari google.com dapat dilihat seperti pada Kode Program 1.

#### Kode Program 1. Contoh Manifest.json

```
1. {
2.   "name": "My Extension",
3.   "version": "2.1",
4.   "description": "Gets information from Google.",
5.   "icons": { "128": "icon_128.png" },
6.   "background": {
7.     "persistent": false,
8.     "scripts": ["bg.js"]
9.   },
10.  "permissions": ["http://*.google.com/", "https://*.google.com/"],
11.  "browser_action": {
12.    "default_title": "",
13.    "default_icon": "icon_19.png",
14.    "default_popup": "popup.html"
15.  }
16. }
```

Arsitektur Google Chrome extension pada umumnya mempunyai sebuah background pages yang berfungsi untuk menyimpan logika utama, sebuah UI (User Interface) sebagai jembatan agar pengguna dapat berinteraksi dengan extension dan juga sebuah content script agar extension dapat berinteraksi dengan web pages.

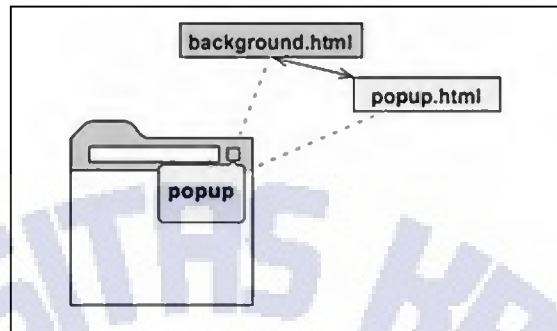


Gambar 1. Arsitektur Background Pages [3]

Gambar 1 merupakan sebuah browser dengan dua extension yang terpasang. Icon dengan warna kuning adalah browser action sedangkan icon dengan warna biru merupakan page action. Kedua action tersebut mempunyai satu background pages yang didefinisikan dalam background.html serta memiliki kode JavaScript yang mengatur perilaku kedua window.

Background pages mempunyai dua tipe pemanggilan yaitu persistent background pages dan even pages. Persistent background pages akan selalu dijalankan pada saat extension aktif sedangkan even pages hanya akan dijalankan pada saat dibutuhkan saja.

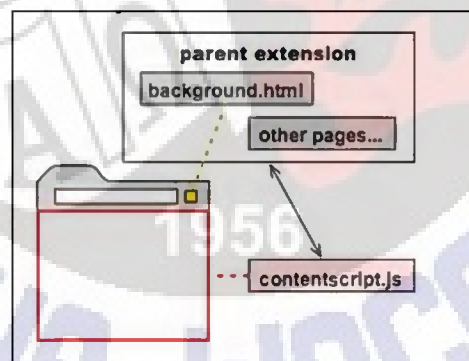
Informasi untuk membedakan kedua tipe tersebut dituliskan pada *file manifest* dari *extension* yang bersangkutan.



Gambar 2. Arsitektur *User Interface Popup* [3]

Gambar 2 merupakan arsitektur dari *UI popup* yang memudahkan *user* untuk berinteraksi dengan *extension*. *Content* dari *popup* tersebut merupakan *web pages* yang didefinisikan dengan *file HTML (popup.html)*. Fungsi yang terdapat pada *background page* juga dapat dipanggil melalui *file popup* tersebut sedangkan penggunaan *popup* harus dideklarasikan terlebih dahulu pada *file manifest*.

Seperti halnya *background pages* dan *UI pages*, *chrome extension* memiliki arsitektur lain berupa *content script*. *Content script* merupakan *file* yang menjembatani agar *extension* dapat berinteraksi dengan *web pages*. Dengan menggunakan teknologi *JavaScript*, sebuah fungsi dalam *content script* akan dijalankan pada saat suatu *web pages* dieksekusi di dalam *browser*. *Content script* ini dapat membaca detail dan melakukan perubahan dari *web pages* tersebut.



Gambar 3. Arsitektur *Content Script* [3]

Gambar 3 menunjukkan bahwa *content script* tidak dapat lepas dari *parent extension*-nya. *Background page* dapat selalu berinteraksi dengan *content script* seperti mengirimkan *message* untuk merubah tampilan halaman pada *browser*.

### **JavaScript Object Notation**

*JavaScript Object Notation (JSON)* merupakan *format* pertukaran data ringan yang mudah dibaca oleh manusia maupun diterjemahkan oleh mesin. *JSON* termasuk bagian dari *JavaScript* akan tetapi *JSON* merupakan bahasa yang dapat berdiri sendiri, sehingga program yang ditulis dengan bahasa pemrograman modern dapat melakukan pertukaran data dengan mudah [4].

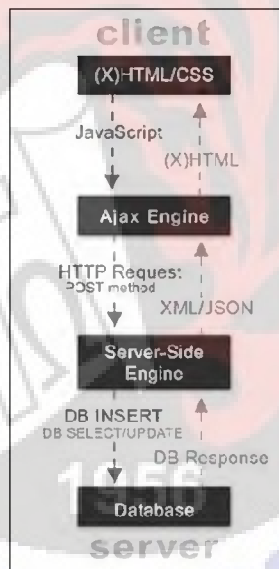


JSON memiliki enam jenis *value*: *object*, *array*, *string*, *number*, *boolean* (*true* dan *false*), dan *null*. *Whitespace* seperti spasi, *tab*, *newline character*, *enter* dan sebagainya dapat dimasukkan ke dalam *format* penulisan JSON sebelum atau sesudah *value* apapun, hal inilah yang membuat JSON lebih mudah dibaca oleh manusia. Sedangkan ketika JSON dieksekusi, *whitespace* tersebut akan dihilangkan sehingga akan menghemat transmisi dan penyimpanan data pada mesin.

### ***XmlHttpRequest***

*XMLHttpRequest* (XHR) adalah sebuah *object JavaScript* yang digunakan untuk mengakses *web server* secara asinkron. XHR akan terus berjalan pada sebuah *web browser*, hal ini terjadi karena proses *request* dilakukan pada *background*. Dengan demikian *refresh browser* tidak diperlukan sepanjang proses *request*. Metode *request* yang biasa digunakan pada XHR adalah GET atau POST yang akan memberikan kemudahan bagi *server* untuk membaca data yang dikirimkan oleh *client* [5].

Terdapat tiga model *request* yang sering digunakan dalam XHR yaitu, *request* dengan *standard XHR*, *request* dengan *Database-Enabled XHR* dan yang ketiga yang digunakan dalam penelitian ini adalah *request* dengan mengirimkan data ke *Database-Enabled XHR*.



**Gambar 4.** Pengiriman Data Ke *Database-Enabled XHR* [5]

**Gambar 4** merupakan *request* yang dilakukan dengan mengirimkan data ke *Database-Enabled XHR*. *Request* di kirimkan melalui XHR GET atau POST ke sebuah bahasa sisi di *server*. Sesudah data diterima oleh *server*, *server* akan memarsing XML atau nilai yang dikirim oleh XHR dan melakukan *query* data ke dalam *database*.

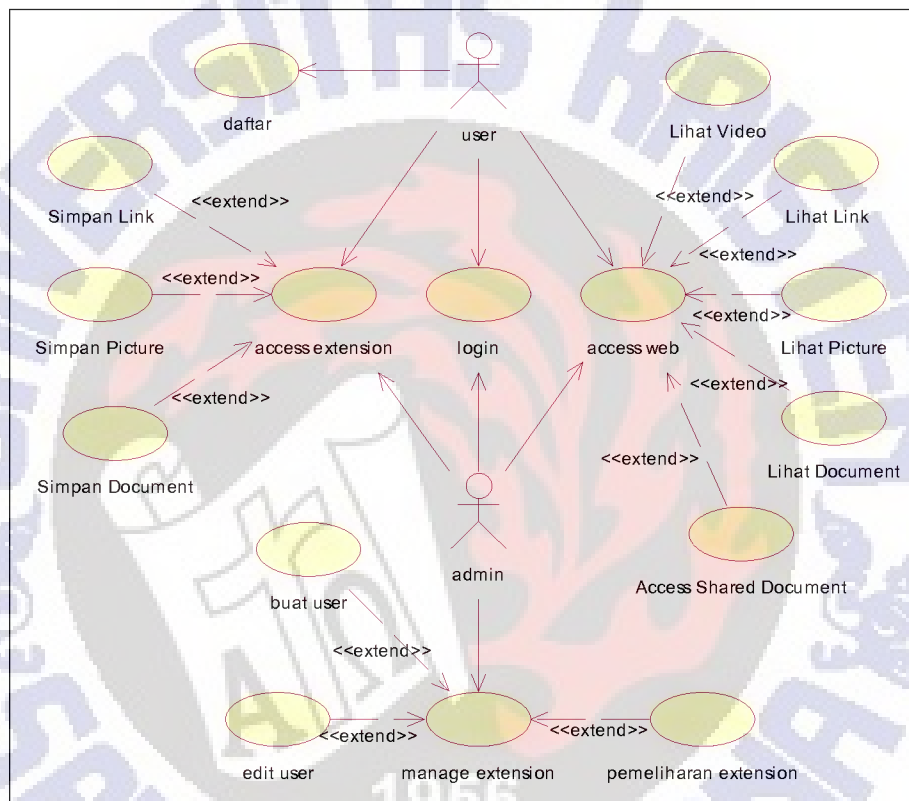
### ***Binary Large Object (Blob)***

*Binary large object* merupakan tipe data di dalam *database* yang dapat menyimpan objek atau data biner. Data biner yang disimpan dalam *database* dapat berupa gambar, *sound file*, *video segment*, atau *file multimedia* yang lain [6]. Dengan menggunakan *blob*, pengelolaan data di dalam *database* dapat dilakukan dengan lebih mudah, pengguna dapat memulihkan atau melakukan *backup database* dengan satu *method* saja sehingga pemberian akses pada *remote clients filesystem* tidak perlu

dilakukan. Penggunaan *blob* juga menjadikan *user* dapat mengakses secara langsung *file* yang tersimpan di dalam *database*.

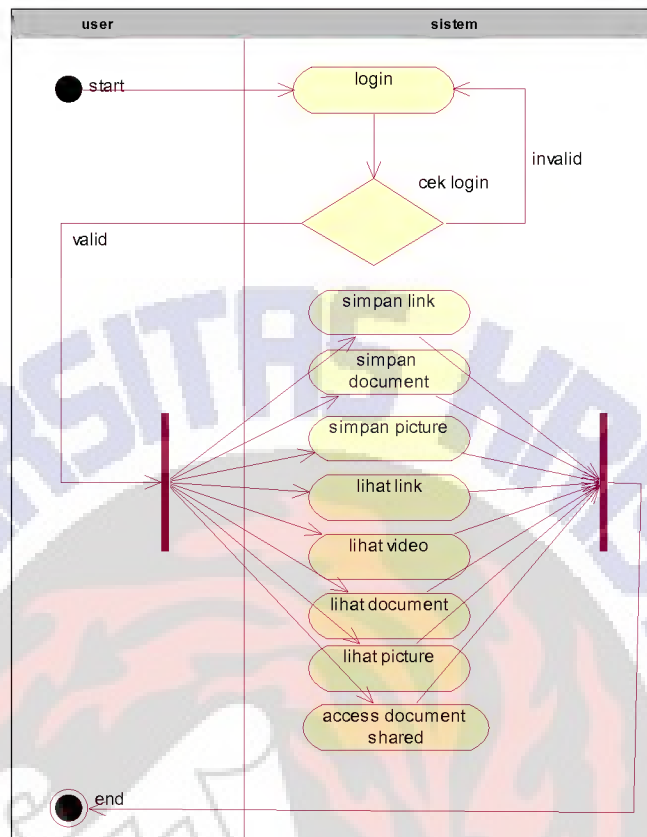
### 3. Perancangan Sistem

Dalam sistem ini, terdapat dua pengguna yaitu, *admin* dan *user*. Untuk masuk ke dalam sistem, *admin* maupun *user* harus *login* terlebih dahulu. *Admin* merupakan *developer* dari *Save to Extension* sedangkan *user* merupakan pengguna yang melakukan pendaftaran ke dalam sistem atau pengguna yang ditambahkan langsung oleh *admin* melalui sistem tersebut.



Gambar 5. Usecase Diagram

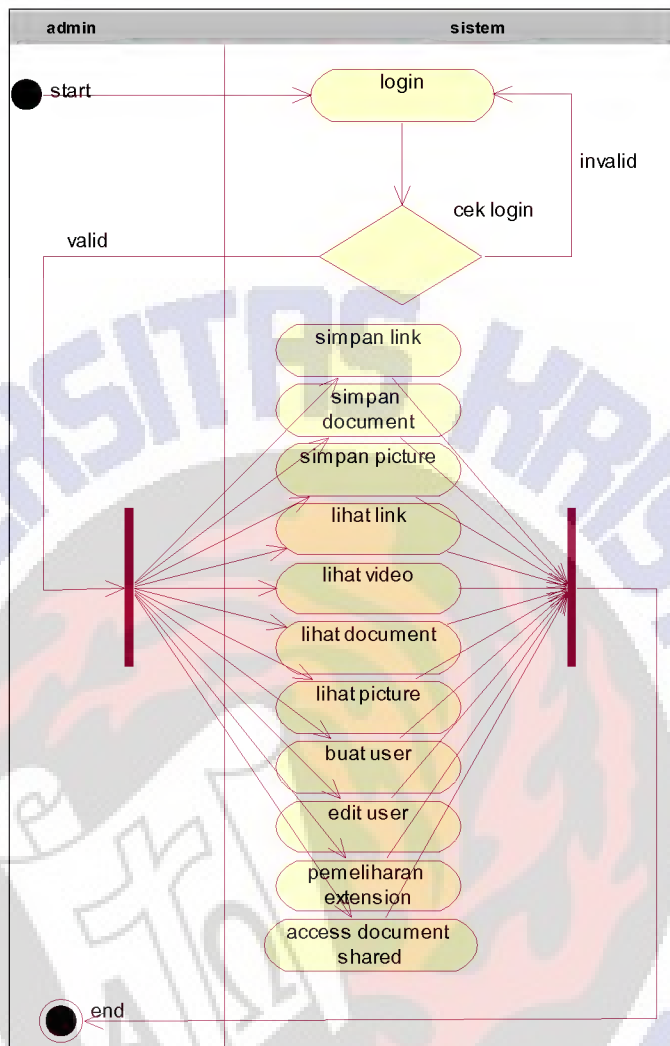
Gambar 5 merupakan *usecase diagram* di dalam pembangunan sistem. Terdapat dua hak akses di dalam *usecase* tersebut, yaitu sebagai *user* dan juga *admin*. *User* dapat mengakses fungsi yang sesuai hak aksesnya setelah melakukan pendaftaran dan *login*, sedangkan *admin* hanya membutuhkan *login* ke dalam sistem untuk mengakses fungsi *admin*.



**Gambar 6.** Activity Diagram User

**Gambar 6** menunjukkan *activity diagram* untuk *user*. *User* harus melakukan *login* terlebih dahulu untuk berinteraksi dengan sistem. Setelah *login*, *user* dapat mengakses beberapa fungsi seperti fungsi "simpan link" yang digunakan untuk menyimpan link URL pada *address bar google chrome*, fungsi "simpan document" untuk menyimpan document (PDF, DOC, XLS, dll.) yang terkandung pada suatu link, fungsi "simpan picture" untuk menyimpan gambar yang ditemui pada *web browser*, fungsi "lihat link" untuk melihat link yang telah disimpan, fungsi "lihat video" untuk melihat link video yang berasal dari *youtube*, fungsi "lihat document" untuk melihat document, fungsi "lihat picture" yang digunakan untuk melihat gambar dan yang terakhir adalah fungsi "access document shared" yaitu fungsi yang dapat digunakan untuk melihat dan mengakses document secara bersama dengan *user* lain.

Seperti halnya *activity diagram* untuk *user*, *admin* harus *login* terlebih dahulu untuk dapat mengakses fungsi-fungsi dari *Save to Extension*. *Admin* memiliki fungsi tambahan yang tidak dapat dilakukan oleh *user*, fungsi tersebut adalah: fungsi "edit user" yang digunakan untuk mengelola serta melakukan *update* informasi *user*, fungsi "buat user" yang digunakan untuk membuat *user* baru dan fungsi "pemeliharaan extension" di mana fungsi ini digunakan untuk mengelola *extension*. *Activity diagram* untuk *admin* dapat dilihat pada **Gambar 7**.



**Gambar 7.** Activity Diagram Admin

Sistem ini dibangun menggunakan 4 (empat) tabel dalam *database*. Tabel-tabel tersebut terdiri dari tabel *save2ext\_user*, *save2ext\_link*, *save2ext\_gambar* dan *save2ext\_file*.

**Tabel *save2ext\_user***

Tabel *save2ext\_user* digunakan untuk menyimpan informasi dari *user* yang terdaftar. *Field* yang terdapat pada tabel ini adalah *field id\_user*, *field username*, *field password*, *field email*, *field level*, *field ip* dan *field firstlog*. Hak akses *user* ditentukan berdasarkan nilai dari *field level*. Secara detail tabel *save2ext* dapat dilihat pada **Tabel 1**.

**Tabel 1.** Tabel *Save2ext\_user*

Field	Type	Keterangan
<b>Id_user</b>	Int(11)	<i>Primary Key</i>
<b>Username</b>	Varchar(32)	<i>Unique</i>
<b>Password</b>	Varchar(32)	
<b>Email</b>	Varchar(255)	<i>Unique</i>
<b>Level</b>	Int(2)	

<b>Ip</b>	Varchar(15)
<b>Firstlog</b>	Datetime

### Tabel Save2ext\_link

Tabel save2ext\_link digunakan untuk menampung data link yang disimpan oleh user. Field yang terdapat pada tabel ini adalah field id, field id\_user, field url dan field title. Secara detail tabel save2ext\_link dapat dilihat pada Tabel 2.

**Tabel 2.** Tabel Save2ext\_link

Field	Type	Keterangan
<b>Id</b>	Int(11)	Primary Key
<b>Id_user</b>	Int(11)	Foreign Key
<b>url</b>	Varchar(1000)	
<b>Title</b>	Varchar(255)	

### Tabel Save2ext\_gambar

Tabel save2ext\_gambar digunakan untuk menampung data gambar yang disimpan oleh user. Field yang terdapat pada tabel ini adalah field id\_gambar, field id\_user, field file\_type, field mime\_type dan field file\_data. Secara detail tabel save2ext\_gambar dapat dilihat pada Tabel 3.

**Tabel 3.** Tabel Save2ext\_gambar

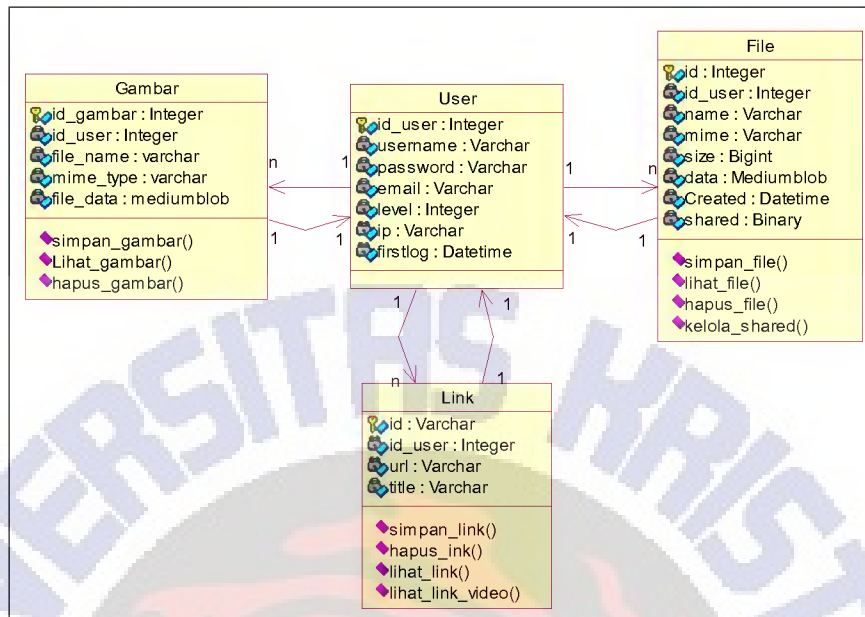
Field	Type	Keterangan
<b>Id_gambar</b>	Int(11)	Primary Key
<b>Id_user</b>	Int(11)	Foreign Key
<b>File_type</b>	Varchar(255)	
<b>Mime_type</b>	Varchar(50)	
<b>File_data</b>	Mediumblob	

### Tabel Save2ext\_file

Tabel save2ext\_file digunakan untuk menampung data document yang disimpan oleh user. Field yang terdapat pada tabel ini adalah field id, field id\_user, field name, field mime, field size, field data, field created dan field shared. Secara detail tabel save2ext\_file dapat dilihat pada Tabel 4.

**Tabel 4.** Tabel Save2ext\_gambar

Field	Type	Keterangan
<b>Id</b>	Int(11)	Primary Key
<b>Id_user</b>	Int(11)	Foreign Key
<b>Name</b>	Varchar(255)	
<b>Mime</b>	Varchar(50)	
<b>Size</b>	Bigint(20)	
<b>Data</b>	Mediumblob	
<b>Created</b>	Datetime	
<b>Shared</b>	Binary(1)	

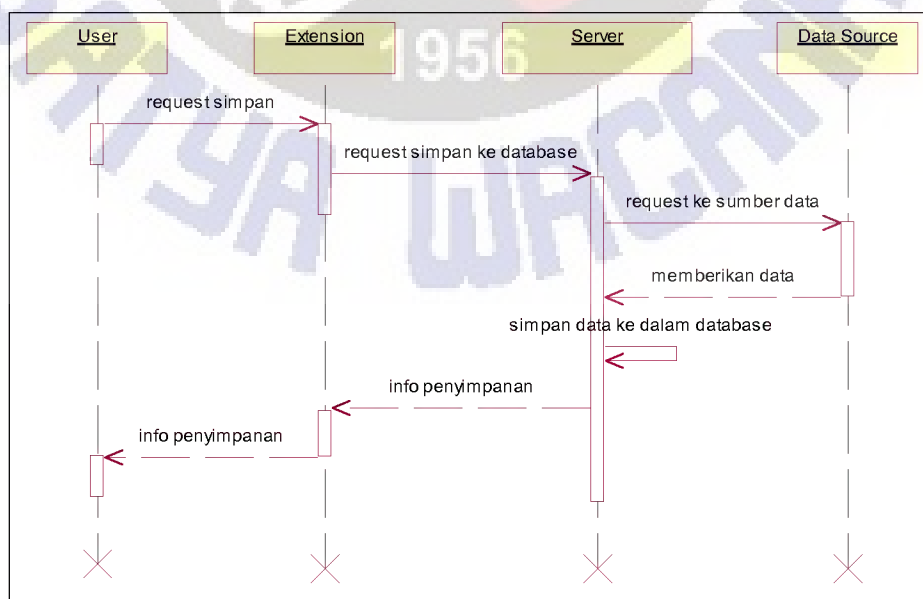


Gambar 8. Class Diagram

Gambar 8 merupakan *class diagram* yang menampilkan relasi antar masing-masing *class*. Pada setiap *class* terdapat *attribute* yang dipakai untuk melakukan *query* ke dalam *database*. *Class* Gambar, Link dan File berhubungan langsung dengan *class* user.

#### 4. Hasil dan Pembahasan

Penelitian ini menghasilkan sebuah aplikasi yang mengintegrasikan antara *chrome extension* dengan *web server*. Dengan menggunakan *interface* yang terdapat pada *web server*, *user* dapat mengelola *file* yang telah tersimpan melalui *chrome extension*. Adapun *interface* yang menjembatani antara *user* dengan sistem terdiri dari, *popup extension*, *context menu*, dan halaman *web*. Proses pertukaran data antara *chrome extension* dan *web server* dapat dilihat pada Gambar 9.



Gambar 9. Proses Pertukaran Data

**Gambar 9** merupakan proses pertukaran data antara *chrome extension* dengan *web server*. Setelah *user* melakukan *request* simpan melalui *extension*, *extension* akan mengirimkan *ajax request* kepada *web server*, *request* tersebut kemudian dilanjutkan oleh *web server* dengan mengunduh data dari *data source*. Data yang diunduh dimasukkan secara langsung ke dalam *database web server*. Setelah data berhasil ditampung dalam *database*, *web server* akan mengirimkan informasi kepada *user* melalui *extension*.

### Popup Extension

Interface dari *popup extension* ditentukan berdasarkan dari *session* yang aktif di dalam *browser*. Proses *cek session* dapat dilihat seperti pada **Gambar 10**.



**Gambar 10.** Proses Request Cek Session

**Gambar 10** merupakan proses *request* *cek session*. Pada saat *load*, *popup extension* akan mengirimkan *ajax request* *cek session* kepada *server*, *server* akan melakukan validasi apakah terdapat *session user* yang tersimpan atau tidak. Apabila tidak ada *session user* yang tersimpan, *server* akan mengirimkan informasi balasan dalam *format JSON* yang kemudian akan diterjemahkan sebagai status *extension*.

### Kode Program 2. Script Ajax Request Cek Session

```

1. $.ajax({
2.     url : 'http://localhost/save2ext/ceksession.php',
3.     data : {ceksession: 'ceksession'},
4.     dataType: 'json',
5.     method: 'post',
6.     success: function(r){
7.         if (r.success){
8.             informasi.text(r.success).fadeIn('fast');
9.             logged.show();
10.        }
11.    }
12.    else if(r.error){
13.        unlogged.show();
14.    }
15. },
16. error: function(r){
17.     informasi.text('Tidak terhubung server').fadeIn('fast');
18. },
19. complete: function(r){
20.     loader.hide();
21. }
22. });
  
```

**Kode Program 2** merupakan *ajax request* yang dikirim oleh *extension* kepada *web server*. Data yang dikirimkan melalui *extension* berupa *variable* *ceksession* dengan string "ceksession". Data tersebut akan diolah melalui *file ceksession.php* di dalam *web server*. Respon yang diterima oleh *extension* akan diterjemahkan untuk menentukan tampilan *popup*. Apabila terdapat *session* yang tersimpan maka *popup* akan menampilkan *form user status*, jika sebaliknya maka *popup* akan menampilkan *form login*.

### Kode Program 3. Script Cek Session Pada Server

```

1. <?php
2. define('INCLUDE_CHECK',true);
3. require 'includes/koneksi.php';
4. session_name('save2ext');
5. session_set_cookie_params(2*7*24*60*60);
6. session_start();
7.
8. function err($error)
9. {
10.     die (json_encode(array('error' => $error)));
11. }
12. $ceklogin = $konek->real_escape_string($_GET['ceksession']);
13. if($ceklogin == 'ceksession'){
14.     if($_SESSION['id']){
15.         echo json_encode(array('success' => 'Anda Login Sebagai
        '.$_SESSION['username']));
16.     }
17.     else
18.     {
19.         err('Anda Belum Login');
20.     }
21. }
22. $konek->close();
23. ?>

```

**Kode Program 3** merupakan *server cek session* yang menampung *request* *cek session* dari *extension*. *Session\_name* pada baris ke-4 (empat) merupakan inisialisasi *session* yang digunakan di dalam sistem. Baris ke-12 merupakan *variable* yang menampung nilai dari data yang dikirimkan melalui *ajax request*. *Server* akan mengirimkan balasan dalam *format* JSON sesuai hasil pengecekan pada baris ke-14.

**Gambar 11.** Tampilan *Popup Login*

**Gambar 11** merupakan *interface* dari *popup login*. *Popup* tersebut ditampilkan jika tidak terdapat *session user* yang aktif. *User* dapat masuk ke dalam sistem dengan

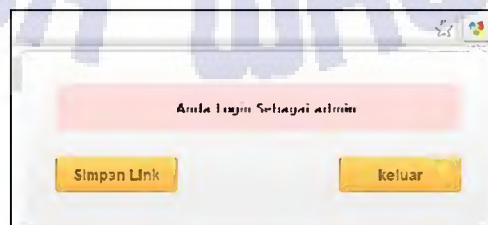


mengisikan *field* *username* dan *password* pada *popup* tersebut. Data yang dimasukan *user* akan dikirim melalui *ajax request* ke *server* cek *login*.

#### Kode Program 4. Script Server Cek Login

```
...
1.  if(!count($err)){
2.
3.      $username = $koneksi->real_escape_string($_GET['username']);
4.      $password = $koneksi->real_escape_string($_GET['password']);
5.      $_GET['rememberMe'] = (int)$_GET['rememberMe'];
6.
7.      $query = "SELECT id_user,username FROM save2ext_user WHERE
      username='$_username' AND password='".md5($password)."'";
8.
9.      $result = $koneksi->query($query);
10.
11.     if ($result->num_rows == 1){
12.
13.         $row = $result->fetch_assoc();
14.
15.         $_SESSION['username']=$row['username'];
16.         $_SESSION['id'] = $row['id_user'];
17.         $_SESSION['rememberMe'] = $_GET['rememberMe'];
18.
19.         setcookie('save2extRemember',$_GET['rememberMe']);
20.
21.         echo json_encode(array('success' => 'Anda Login Sebagai
22.         '.$_SESSION['username']));
23.     }
24.     else
25.     {
26.         err('Username / Password Salah !');
27.     }
28. }
...
```

Kode Program 4 merupakan *server* cek *login* yang digunakan untuk menampung *request* cek *login* dari *extension*. Data yang dikirimkan melalui *ajax request* ditampung pada *variable* *username* dan *password*. Nilai dari *variable* tersebut kemudian dicocokkan dengan nilai yang terdapat pada *database*. Jika data sesuai maka sistem akan menyimpan *session id* dan nama *user* di dalam sistem kemudian mengubah *interface popup* menjadi seperti pada Gambar 12.



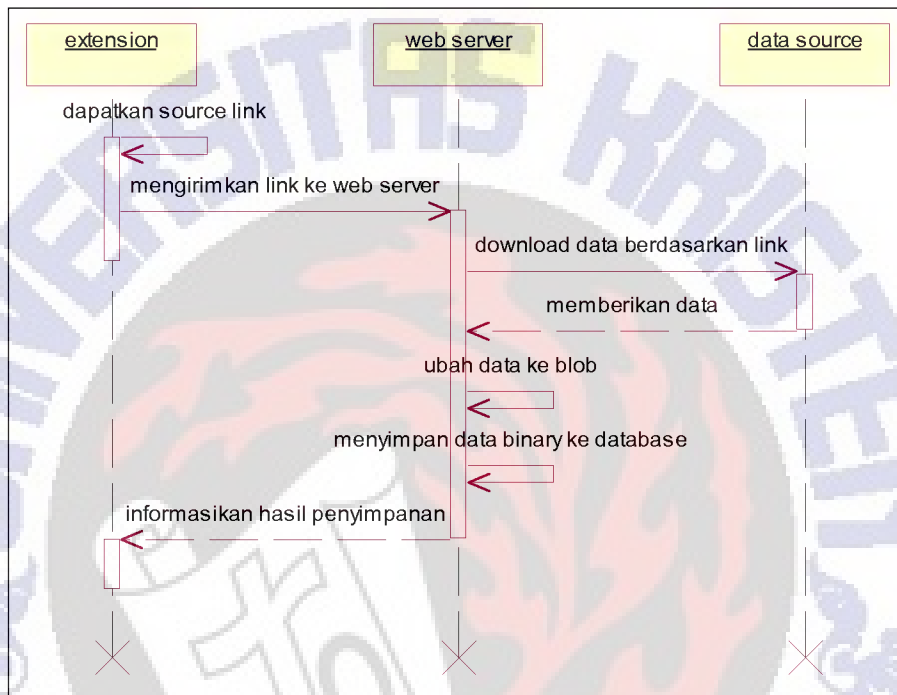
Gambar 12. Tampilan *Popup* User Status

Gambar 12 merupakan *popup* yang ditampilkan jika terdapat *session user* yang aktif. Melalui *popup* tersebut *user* dapat menyimpan *link* URL pada halaman yang terbuka. Penyimpanan *link* melalui tombol “Simpan Link” dilakukan dengan

mengirimkan *ajax request* ke *server*. *Ajax request* tersebut kemudian ditampilkan dan dikelola di dalam *server upload link*.

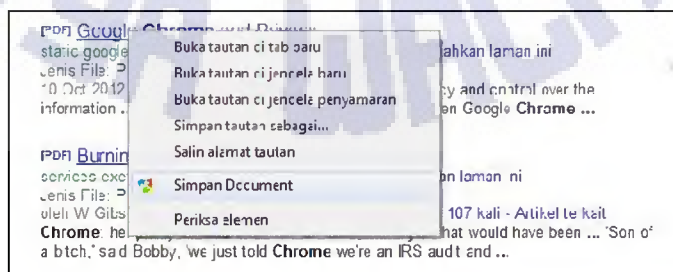
### Context Menu

*Extension* yang dibangun ini menampilkan *context menu* pada halaman *chrome browser*. *Context menu* ini berfungsi untuk menangani *object* yang berupa gambar dan *link*.



Gambar 13. Proses Penyimpanan Data

**Gambar 13** merupakan proses penyimpanan data ke dalam *database web server*. Informasi *link* yang didapatkan melalui *context menu*, dikirimkan melalui *XmlHttpRequest* kepada *web server*. *Web server* kemudian akan melakukan *download* data berdasarkan *link* tersebut, mengubah ke dalam bentuk *blob* (*binary large object*) dan menyimpannya di dalam *database*.



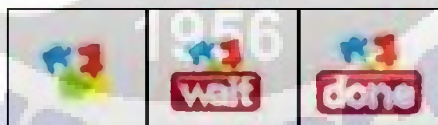
Gambar 14. Tampilan Context Menu Pada Object Link

*Object* yang berupa *link* pada *web browser* apabila mendapatkan interaksi dari *user* berupa “klik kanan” akan menghasilkan *context menu* seperti pada **Gambar 14**. Menu “Simpan Document” pada *context menu* akan memanggil fungsi *upload file* dari *background page*. Fungsi *upload file* dapat dilihat pada **Kode Program 5**.

### Kode Program 5. Script Fungsi Upload File

```
1. function uploadFile(info) {
2.   chrome.browserAction.setBadgeText ( { text: "wait" } );
3.   var x = new XMLHttpRequest();
4.   x.onload = function() {
5.     var link = info.linkUrl.split(/[?#]/)[0].split('/').pop();
6.     var fd = new FormData();
7.     fd.append("uploaded_file", x.response, link);
8.     var y = new XMLHttpRequest();
9.     y.open('POST', 'http://localhost/save2ext/includes/upload_file.php');
10.    y.onreadystatechange = function() {
11.      if (y.readyState == 4) {
12.        if (y.status == 200) {
13.          chrome.browserAction.setBadgeText ( { text: "done" } );
14.          setTimeout(function () {
15.            chrome.browserAction.setBadgeText( { text: "" } );
16.          }, 1000);
17.          notifikasi();
18.        }
19.      }
20.    };
21.    y.send(fd);
22.  };
23.  x.responseType = 'blob';
24.  x.open('GET', info.linkUrl);
25.  x.send();
26. }
```

**Kode Program 5** merupakan fungsi *upload file* yang akan mengambil informasi suatu *file* dari *link*. Informasi tersebut dikirim ke *server* melalui *XmlHttpRequest*. Melalui *server* informasi tersebut diubah menjadi bentuk *blob* dan disimpan di dalam *database*. Saat proses penyimpanan pada *server* masih berjalan, *extension* akan menampilkan *badge* pada *icon extension*. *Badge* tersebut dideklarasikan dengan menggunakan *script chrome.browserAction.setBadgeText*. *Badge* akan berubah dan menampilkan informasi selesai jika proses penyimpanan berhasil.



**Gambar 15.** Tampilan *Badge* Pada *Icon*

**Gambar 15** merupakan perubahan *badge icon* dari tampilan biasa, sedang terjadi proses penyimpanan di dalam *server*, dan yang terakhir saat proses penyimpanan telah selesai.



**Gambar 16.** Tampilan *Context Menu* Pada *Object Gambar*

*Object* gambar akan menampilkan menu “simpan gambar” pada saat *user* melakukan interaksi berupa “klik kanan” seperti pada **Gambar 16**. Menu “simpan gambar” akan memanggil fungsi *upload* gambar dari *background page*. Fungsi tersebut dapat dilihat pada **Kode Program 6**.

**Kode Program 6. Script Fungsi Upload Gambar**

```

1. function uploadGambar(info) {
2.     chrome.browserAction.setBadgeText ( { text: "wait" } );
3.     var x = new XMLHttpRequest();
4.     x.onload = function() {
5.         var file_name = info.srcUrl.split(/[?#]/)[0].split('/').pop();
6.         var fd = new FormData();
7.         fd.append("gambar", x.response, file_name);
8.         var y = new XMLHttpRequest();
9.         y.open('POST', 'http://localhost/save2ext/includes/upload_gambar.php');
10.        y.onreadystatechange = function() {
11.            if (y.readyState == 4) {
12.                if (y.status == 200) {
13.                    chrome.browserAction.setBadgeText ( { text: "done" } );
14.                    setTimeout(function () {
15.                        chrome.browserAction.setBadgeText( { text: "" } );
16.                    }, 1000);
17.                    notifikasi();
18.                }
19.            }
20.        };
21.        y.send(fd);
22.    };
23.    x.responseType = 'blob';
24.    x.open('GET', info.srcUrl);
25.    x.send();
26. }

```

**Kode Program 6** merupakan kode program yang digunakan untuk mengambil informasi dari suatu gambar, dan mengirimkan informasi dari gambar tersebut ke *server* dengan menggunakan *XmlHttpRequest*. Data yang dikirimkan diubah pada *server* ke dalam bentuk *blob* dan disimpan ke dalam *database*.

**Halaman Web**

Halaman *web* digunakan oleh *user* untuk mengelola kembali *file* yang telah disimpan melalui *chrome extension*. Halaman tersebut terdiri dari halaman *My Picture*, halaman *My Links*, halaman *My Video*, halaman *My Document* dan halaman *Shared Document*. Setiap *document* yang di *share* pada halaman *My Document* dapat diakses oleh semua *user* melalui halaman *Shared Document*.

Owner	Name	Mime	Size (bytes)	
admin	google-chrome-privacy-whitepaper.pdf	application/pdf	319288	<a href="#">Download</a>
dulharjo	google-chrome-privacy-whitepaper.pdf	text/html	64005	<a href="#">Download</a>

**Gambar 17.** Tampilan Halaman *Shared Document*

**Gambar 17** merupakan tampilan dari halaman *shared document*. Melalui halaman ini *user* dapat mengelola *file* yang akan dibagikan dengan *user* lain. Fungsi *shared document* ditentukan berdasarkan nilai dari *field level* untuk setiap *record* dari tabel *Save2ext\_file*. Apabila nilai dari *field level* tersebut sama dengan 1 (satu) maka *record* yang bersangkutan dapat diakses oleh *user* yang lain namun apabila nilai *field level* sama dengan 0 (nol) maka *file* tersebut hanya dapat diakses oleh *user* yang bersangkutan. Nilai dari *field level* secara *default* adalah 0 (nol), nilai ini ditentukan dari dalam *database* secara langsung. Ketika *user* melakukan *request* simpan *document* melalui *XMLHttpRequest*, maka *database* pada *server* akan secara otomatis menambahkan nilai 0 (nol) pada *field shared*. Nilai *field shared* dapat diubah melalui halaman *my document*. Pada halaman tersebut terdapat *icon* yang menandakan bahwa *document* yang bersangkutan dapat diakses oleh *user* lain atau tidak. *Script* yang digunakan untuk menangani fungsi tersebut dapat dilihat pada **Kode Program 7**.

**Kode Program 7. Script Untuk Menampilkan Icon Shared**

```

1.  if($row['shared']=='0'){
2.      echo "<a href=includes/update_shared.php?id=".$row['id']. "><img
   src=images/noshared.png></img></a>";
3.  }
4.  else{
5.      echo "<a href=includes/update_shared.php?id=".$row['id']. "><img
   src=images/shared.png></img></a>";
6.  }

```

**Kode Program 7** merupakan *script* yang digunakan untuk menentukan *icon* pada halaman *my document*. Apabila di dalam *database field shared* bernilai 0 maka yang akan ditampilkan pada halaman *my document* adalah *icon* silang sedangkan jika *field shared* bernilai 1 maka *icon* yang ditampilkan adalah *icon* centang. *Icon* tersebut diarahkan pada *script update\_shared.php*, di mana kode program ini akan mengubah hak akses dari *document* yang bersangkutan.

**Kode Program 8. Script Untuk Mengubah Nilai Shared**

```

1.  while($row = $result->fetch_assoc()) {
2.      if($row['shared']=='0'){
3.          $query2="UPDATE save2ext_file SET shared=1 WHERE id='$id'";
4.      }
5.      else
6.      {
7.          $query2="UPDATE save2ext_file SET shared=0 WHERE id='$id'";
8.      }
9.  }

```

**Kode Program 8** digunakan untuk mengubah nilai dari *field Shared* dengan melakukan *query update* ke dalam *database Save2ext\_file*. Setelah fungsi ini dijalankan maka *document* yang bersangkutan akan masuk pada *document shared* di dalam *website*.

**5. Pengujian Sistem**

Pengujian sistem dilakukan dengan 2 (dua) tahap. Tahap pertama merupakan *alpha testing* yang dilakukan dengan metode *black box*. Pengujian kedua merupakan *beta testing* yang dilakukan dengan memberikan kuisioner kepada 30 orang responden. *Alpha testing* dapat dilihat seperti pada **Tabel 5**.

**Tabel 5. Black Box Testing**

Jenis uji	Nilai uji	Output yang diharapkan	Keterangan
Uji Login	Benar	- Tidak ada pesan error - Session tersimpan - User dapat menyimpan data melalui <i>extension</i>	Diterima
	Salah	- Menimbulkan pesan <i>error</i> - Session tidak tersimpan - User tidak dapat menyimpan data melalui <i>extension</i> .	Diterima
Uji Pendaftaran	Benar	- Mengirim informasi <i>login</i> melalui <i>email</i>	Diterima
	Salah	- Menimbulkan pesan <i>error</i>	Diterima
Simpan Link	Benar	- Link tersimpan di dalam <i>database web server</i> . - Tidak ada pesan <i>error</i>	Diterima
	Salah	- Link tidak tersimpan - Menimbulkan pesan <i>error</i>	Diterima
Simpan Gambar	Benar	- Gambar tersimpan di dalam <i>database web server</i> - Menampilkan notifikasi dengan benar	Diterima
	Salah	- Gambar tidak tersimpan dengan benar	Diterima
Simpan Document	Benar	- Document tersimpan di dalam <i>database</i> - Menampilkan notifikasi dengan benar	Diterima
	Salah	- Document tidak tersimpan dengan benar	Diterima

Tabel 5 Merupakan hasil dari pengujian *black box testing*. Secara umum Berdasarkan nilai uji yang diberikan, aplikasi yang dibangun dapat berjalan seperti yang telah diharapkan.

Berdasarkan Penelitian yang dilakukan dengan cara kuisioner kepada pengguna *internet*, diperoleh hasil bahwa: 1) 43% responden belum pernah menggunakan aplikasi *chrome extension* sebelumnya, 40% pernah menggunakan dan 17% sering menggunakan, 2) 70% responden berpendapat bahwa aplikasi ini memudahkan dalam melakukan *download file*, 17% berpendapat sangat memudahkan dan 13% berpendapat tidak memudahkan, 3) 57% responden berpendapat bahwa *extension* ini memudahkan di dalam berbagi *file*, 23% berpendapat sangat memudahkan dan 20% tidak memudahkan, 4) 60% responden berpendapat bahwa aplikasi ini membantu di dalam menggunakan *file* kembali, 23% berpendapat sangat membantu sedangkan 17% berpendapat tidak membantu, 5) 73% responden berpendapat bahwa aplikasi ini menghemat kapasitas *hardisk* sedangkan 27% responden berpendapat sangat membantu, 6) 56% responden berpendapat bahwa aplikasi ini membantu di dalam mengelola *file* dengan lebih baik, 27% berpendapat sangat membantu sedangkan 17% berpendapat tidak membantu, 7) Menurut 75% responden aplikasi ini menarik, 11% berpendapat sangat menarik sedangkan 14% berpendapat tidak menarik.

## 6. Simpulan

Berdasarkan pembahasan pada bab-bab sebelumnya, maka dapat diambil kesimpulan bahwa pembangunan aplikasi ini berguna bagi *user*. Aplikasi ini dapat

mengintegrasikan *file download* di dalam *web server* dengan baik, karena *link*, gambar, dan *document* yang telah disimpan melalui *re-host* dapat digunakan kembali dengan mudah. Keuntungan lain bagi *user* adalah *user* dapat berbagi *file document* bersama-sama dengan *user* yang lain.

## 7. Daftar Pustaka

- [1] *Browser Statistics and Trends*.  
[http://www.w3schools.com/browsers/browsers\\_stats.asp](http://www.w3schools.com/browsers/browsers_stats.asp). (diakses tanggal 16 Juli 2012).
- [2] Kitada, Christina, 2011. *Perancangan dan Implementasi Music Downloader Extension Dengan Platform Google Chrome*. Salatiga: Fakultas Teknologi Informasi UKSW.
- [3] *Google Chrome Extension Overview*.  
<https://developer.chrome.com/extensions/overview.html>. (diakses tanggal 16 Juli 2012).
- [4] Crockford, Douglas, 2008. *Javascript: The Good Parts*. O'reilly, 136-139.
- [5] Hadlock, Kris. 2007. *Ajax for Web Application Developers*. Sams Publishing, USA.
- [6] Anonymous, 1999. *Interbase 6 Data Definition Guide*. Borland/Inprise. CA.

