

**RANCANG BANGUN APLIKASI *SEARCH ENGINE REPOSITORY*  
*SOFTWARE DESKTOP* MENGGUNAKAN TEKNIK *WEB EXTRACTION***

**(SKRIPSI)**

**Oleh**

**DAKO ADI AHMADI**



**JURUSAN ILMU KOMPUTER  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDARLAMPUNG  
2016**

## **ABSTRACT**

### ***THE DESIGN AND APPLICATIONS OF SEARCH ENGINE REPOSITORY SOFTWARE DESKTOP USING WEB EXTRACTION TECHNIQUE***

**By**

**Dako Adi Ahmadi**

*The System Search Engine Repository Software Desktop was created with the aim for facilitating the users to get the software to be installed or used by the user. For Windows operating systems and Macintosh, the system can download and check the latest version of the software that is already available. The system created provides a freeware software for Windows and Macintosh operating systems. For Linux, the system displays and generates sources.list of Debian. In this research the system made using Extreme Programming and using web extraction techniques to extract data from website [www.filehippo.com](http://www.filehippo.com).*

**Keywords:** *web scraping, search engine, extreme programming, web extraction, repository software.*

## ABSTRAK

### **RANCANG BANGUN APLIKASI *SEARCH ENGINE REPOSITORY SOFTWARE DESKTOP* MENGGUNAKAN TEKNIK *WEB EXTRACTION***

Oleh

**Dako Adi Ahmadi**

Sistem *Search Engine Repository Software Desktop* dibuat dengan tujuan mempermudah pengguna untuk mendapatkan *software* yang akan di-*install* atau digunakan oleh pengguna. Untuk *software* sistem operasi Windows dan Macintosh sistem dapat mengunduh dan mengecek versi terakhir dari *software* yang telah tersedia. Sistem yang dibuat menyediakan *software* yang bersifat *freeware* untuk sistem operasi Windows dan Macintosh. Pada menu Linux, sistem menampilkan dan melakukan *generate sources.list* dari distro Debian. Pada penelitian ini sistem yang dibuat menggunakan metode *Extreme Programming* dan menggunakan teknik *web extraction* untuk mengambil data dari situs [www.filehippo.com](http://www.filehippo.com).

**Kata Kunci:** *web scraping, search engine, extreme programming, web extraction, repository software*

**RANCANG BANGUN APLIKASI *SEARCH ENGINE REPOSITORY*  
*SOFTWARE DESKTOP* MENGGUNAKAN TEKNIK *WEB EXTRACTION***

Oleh

**DAKO ADI AHMADI**

**Skripsi**

Sebagai Salah Satu Syarat untuk Memperoleh Gelar  
**SARJANA KOMPUTER**

Pada

**Jurusan Ilmu Komputer  
Fakultas Matematika dan Ilmu Pengetahuan Alam**



**JURUSAN ILMU KOMPUTER  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDARLAMPUNG  
2016**

## PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul "Rancang Bangun *Aplikasi Search Engine Repository Software Desktop Menggunakan Teknik Web Extraction*" merupakan karya saya sendiri dan bukan hasil karya orang lain. Semua hasil tulisan yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi saya ini merupakan hasil duplikasi atau dibuat oleh orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, Juni 2016



Dako Adi Ahmadi  
NPM. 0917032030

Judul Skripsi : **RANCANG BANGUN APLIKASI *SEARCH ENGINE REPOSITORY SOFTWARE* DESKTOP MENGGUNAKAN TEKNIK *WEB EXTRACTION***

Nama Mahasiswa : **Dako Adi Ahmadi**

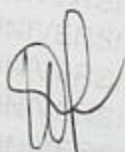
Nomor Pokok Mahasiswa : 0917032030

Jurusan : Ilmu Komputer

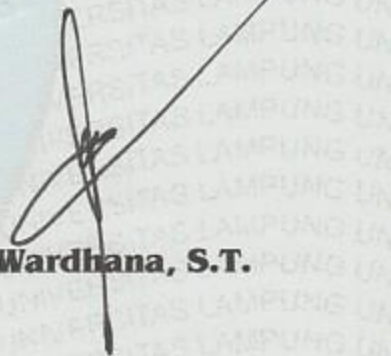
Fakultas : Matematika dan Ilmu Pengetahuan Alam

**MENYETUJUI**

**1. Komisi Pembimbing**

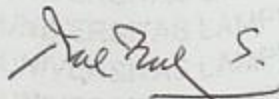


**Dra. Wamilliana, M.A., Ph.D.**  
NIP 19631108 198902 2 001



**Wisnu Wardhana, S.T.**

**2. Ketua Jurusan Ilmu Komputer**



**Dr. Ir. Kurnia Muludi, M.S.Sc.**  
NIP 19640616 198902 1 001

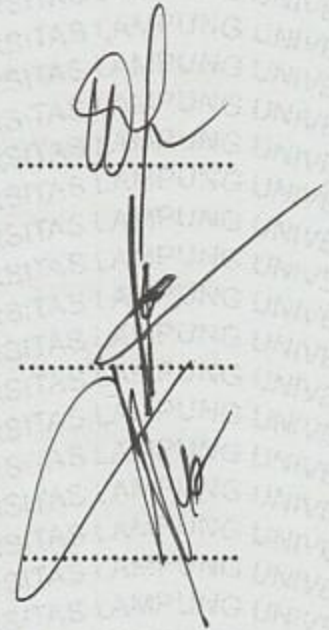
## MENGESAHKAN

### I. Tim Penguji

Ketua : **Dra. Wamiliana, M.A., Ph.D.** .....

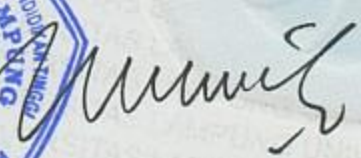
Sekretaris : **Wisnu Wardhana, S.T.** .....

Penguji  
Bukan Pembimbing : **Didik Kurniawan, S.Si., M.T.** .....



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



  
**Prof. Warsito, S.Si., D.E.A.**  
NIP 19710212 199512 1 001

Tanggal Lulus Ujian Skripsi : **29 April 2016**

## **RIWAYAT HIDUP**

Penulis dilahirkan pada tanggal 18 Maret 1991 di Kabupaten Tulang bawang Provinsi Lampung. Penulis merupakan anak pertama dari dua bersaudara yang dilahirkan dari pasangan suami istri yang bernama Pitono dan Murtini.

Penulis memulai pendidikannya di Sekolah Dasar (SD) di SDN 1 Lesung Bakti Jaya, kabupaten Tulang Bawang Barat tahun 1997. Setelah itu, penulis melanjutkan pendidikan Sekolah Menengah Pertama (SMP) pada tahun 2003 di SMPN 1 Lambu Kibang, Tulang Bawang Barat. Pada tahun 2006 penulis melanjutkan pendidikan di SMAN 1 Banjar Agung, Tulang Bawang.

Setelah lulus SMA pada tahun 2009, penulis melanjutkan pendidikan di Universitas Lampung melalui jalur Seleksi Nasional Masuk Perguruan Tinggi Negeri (SNMPTN). Penulis memilih Program Studi Ilmu Komputer, Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam.

Selama berkuliah di Universitas Lampung, penulis tidak hanya aktif dalam kegiatan perkuliahan, tetapi penulis juga aktif berorganisasi dalam kampus.

Berikut ini adalah daftar riwayat organisasi penulis:

1. Anggota Muda Rois FMIPA Unila
2. Anggota Muda Himatika Unila
3. Anggota Muda BEM FMIPA Unila



4. Kepala Bidang INFOKOMEDIA HIMAKOM Unila

Pada bulan Januari 2013 sampai Februari 2013, penulis melakukan kerja praktik di Dinas Bina Marga Bandar Lampung.

## **MOTTO**

*Gagal adalah ketika kita berhenti mencoba melakukan sesuatu. Selalu ada peluang untuk berhasil selama terus mencoba.*

*Perjalanan sejauh 1000 Mil dimulai dari langkah yang pertama*

## **PERSEMBAHAN**

Puji dan syukur kehadiran Allah SWT  
dengan segala kerendahan hati saya persembahkan  
karya ini untuk:

Ayah dan ibu saya yang selalu  
memberikan saya dukungan materil dan moril  
sehingga saya dapat menyelesaikan karya ini dengan baik

Adik kandung saya tercinta

Almamater tercinta Universitas Lampung

dan seluruh sahabat saya  
baik sahabat yang ada dalam almamater tercinta  
atau sahabat yang berada di luar almamater yang telah  
menjadi bagian dalam proses hidup dan kehidupan

## SANWACANA

Puji Syukur kehadiran Allah SWT, atas berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan skripsi. Sholawat serta salam semoga senantiasa kita sanjungkan kepada nabi kita yang selalu sabar dan tak pernah putus asa untuk menegakkan agama islam yaitu Rasulullah Muhammad SAW beserta keluarga dan para sahabat. Mudah-mudahan kita mendapat pertolongan beliau di hari kiamat nanti.

Skripsi ini disusun sebagai syarat untuk memperoleh gelar Sarjana Komputer di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung dan berjudul “Rancang Bangun *Aplikasi Search Engine Repository Software Desktop* Menggunakan Teknik *Web Extraction*”.

Penelitian ini tidak akan selesai bila tidak didukung oleh orang-orang yang selalu memberikan bimbingan, motivasi dan semangat untuk penulis. Untuk itu penulis ingin mengucapkan terima kasih kepada orang-orang yang telah berjasa dalam penyusunan skripsi ini. Berikut ini adalah orang-orang yang berjasa dan pantas mendapatkan ucapan terima kasih dari penulis:

1. Ayah dan ibu serta adik saya yang telah memberikan dukungan, baik itu dukungan materil dan maupun non materil.
2. Ibu Dra. Wamiliana, M.A., Ph.D., sebagai pembimbing I yang telah

memberikan masukan dalam pengerjaan skripsi ini ditengah-tengah kesibukannya.

3. Bapak Wisnu Wardhana, S.T., sebagai pembimbing II penulis dan pimpinan Linux Lampung.
4. Bapak Didik Kurniawan, S.Si., M.T., selaku pembahas, yang berkenan memberikan kritik dan saran dalam skripsi ini.
5. Bapak Dr. Ir. Kurnia Muludi M.S.Sc., selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
6. Seluruh dosen jurusan Ilmu Komputer Universitas Lampung.
7. Teman-teman jurusan Ilmu Komputer Universitas Lampung angkatan 2009 yang selalu memberikan semangat dan motivasi kepada penulis.
8. Semua orang yang telah menjadi inspirasi hidup.

Penulis menyadari bahwa skripsi ini masih terdapat banyak kekurangan. Oleh karena itu penulis sangat mengharapkan kritik dan saran yang membangun dari pembaca, dan semoga skripsi ini dapat bermanfaat bagi yang membacanya.

Bandar Lampung, Juni 2016

Dako Adi Ahmadi

## DAFTAR ISI

	<b>Halaman</b>
<b>SANWACANA .....</b>	<b>x</b>
<b>DAFTAR ISI.....</b>	<b>xii</b>
<b>DAFTAR TABEL .....</b>	<b>xiv</b>
<b>DAFTAR GAMBAR.....</b>	<b>xv</b>
 <b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	4
 <b>BAB II TINJAUAN PUSTAKA</b>	
2.1 <i>Web Scraping</i> .....	5
2.2 <i>Search Engine</i> .....	7
2.3 <i>Pyramid Framework</i> .....	7
2.4 <i>DNS (Domain Name Service)</i> .....	9
2.5 Metode Pengembangan Aplikasi.....	9
2.6 <i>Black Box Testing</i> .....	12
 <b>BAB III METODE PENELITIAN</b>	
3.1 Tempat dan Waktu Penelitian .....	14
3.2 Alat Pendukung .....	14
3.2.1 <i>Hardware</i> .....	14
3.2.2 <i>Software</i> .....	15
3.2.2.1 <i>Sistem Operasi</i> .....	15
3.2.2.2 <i>Software Aplikasi</i> .....	15
3.3 Metode Pengembangan Sistem.....	16

3.3.1 <i>Planning</i> (Perencanaan).....	16
3.3.2 Design (Desain) .....	16
3.3.3 <i>Coding</i> (Pengkodean) .....	28
3.3.4 <i>Testing</i> (Pengujian Sistem) .....	28

#### **BAB IV PEMBAHASAN**

4.1 Instalasi Pyramid.....	30
4.2 Instalasi AngularJS .....	34
4.3 <i>Setting</i> DNS .....	38
4.4 Tampilan Sistem .....	45
4.5 Hasil Pengujian .....	53

<b>SIMPULAN DAN SARAN .....</b>	<b>56</b>
---------------------------------	-----------

<b>DAFTAR PUSTAKA .....</b>	<b>57</b>
-----------------------------	-----------

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Proses <i>Extreme Programming</i> .....	11
Gambar 3.1 <i>Mockup</i> Halaman <i>Home</i> .....	17
Gambar 3.2 <i>Mockup</i> Halaman <i>Windows</i> .....	18
Gambar 3.3 <i>Mockup</i> Halaman <i>Macintosh</i> .....	19
Gambar 3.4 <i>Mockup</i> Halaman <i>Linux</i> .....	20
Gambar 3.5 <i>Mockup</i> Halaman <i>Generate Sources.list</i> .....	20
Gambar 3.6 <i>Mockup</i> Halaman Pencarian Berdasarkan Kategori.....	21
Gambar 3.7 <i>Mockup</i> Pencarian Berdasarkan Kata Kunci.....	22
Gambar 3.8 <i>Mockup</i> Halaman <i>Download</i> .....	23
Gambar 3.9 <i>Sequence Diagram</i> Halaman <i>Windows</i> .....	24
Gambar 3.10 <i>Sequence Diagram</i> Halaman <i>Macintosh</i> .....	25
Gambar 3.11 <i>Sequence Diagram</i> Pencarian <i>Software Windows</i> .....	26
Gambar 3.12 <i>Sequence Diagram</i> Pencarian <i>Software Macintosh</i> .....	27
Gambar 3.13 <i>Sequence Diagram</i> <i>Debian Generator</i> .....	28
Gambar 4.1 Instalasi <i>Dependency Python</i> .....	30
Gambar 4.2 <i>Decompression Python</i> .....	31
Gambar 4.3 Konfigurasi Instalasi <i>Python</i> .....	31
Gambar 4.4 Perintah <i>Make</i> .....	32
Gambar 4.5 Perintah <i>Make Install</i> .....	32
Gambar 4.6 Instalasi <i>Build Essential</i> .....	33
Gambar 4.7 Instalasi <i>Virtual Environment</i> .....	33
Gambar 4.8 <i>Generate Folder Venv</i> .....	34
Gambar 4.9 Instalasi <i>Pyramid</i> .....	34
Gambar 4.10 Instalasi <i>NodeJS</i> .....	35



Gambar 4.11 Instalasi Curl .....	35
Gambar 4.12 Instalasi Bower dan Grunt.....	35
Gambar 4.13 Instalasi Grunt-cli.....	36
Gambar 4.14 Instalasi Yoeman.....	36
Gambar 4.15 Instalasi Bootstrap Yo Angular.....	37
Gambar 4.16 Menjalankan <i>Grunt Serve</i> .....	37
Gambar 4.17 Instalasi AngularJS Berhasil .....	38
Gambar 4.18 Pengaturan <i>IP Address</i> .....	39
Gambar 4.19 Membuat <i>File Reverse</i> .....	39
Gambar 4.20 Membuat File Forward.....	40
Gambar 4.21 Melakukan <i>Edit File</i> db.repo.....	40
Gambar 4.22 <i>File</i> db.repo Belum Dikonfigurasi .....	40
Gambar 4.23 <i>File</i> db.repo Setelah Dikonfigurasi .....	41
Gambar 4.24 <i>File</i> db.rev Belum Dikonfigurasi .....	41
Gambar 4.25 Konfigurasi <i>File</i> db.rev .....	42
Gambar 4.26 Konfigurasi <i>named.local.conf</i> .....	43
Gambar 4.27 Proses <i>Restart Service Networking</i> .....	43
Gambar 4.28 Testing Konfigurasi DNS.....	44
Gambar 4.29 <i>Setting File</i> resolv.conf .....	44
Gambar 4.30 <i>Restart Bind Service</i> .....	45
Gambar 4.31 Aktivasi <i>Service Pyramid</i> .....	45
Gambar 4.32 Membuka Program Aplikasi .....	46
Gambar 4.33 Halaman Awal.....	46
Gambar 4.34 Halaman Windows .....	47
Gambar 4.35 Halaman Macintosh.....	48
Gambar 4.36 Halaman Linux.....	49
Gambar 4.37 <i>Alert Error</i> .....	49
Gambar 4.38 Halaman <i>Generate Sources.list</i> .....	50
Gambar 4.39 Halaman <i>Search</i> .....	51
Gambar 4.40 Halaman <i>Search</i> Berdasarkan Kategori .....	51
Gambar 4.41 Halaman <i>Download</i> .....	52
Gambar 4.42 Halaman Konfirmasi <i>Download</i> .....	52

Gambar 4.43 Halaman *About* ..... 53

## DAFTAR TABEL

	Halaman
Tabel 1. Hasil Pengujian. ....	53

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Saat ini teknologi berkembang dengan sangat pesat, terutama perkembangan dunia komputer. Perkembangan itu baik dalam *perkembangan hardware* ataupun perkembangan *software*. *Software* dan *hardware* memiliki peranan masing-masing yang saling terkait. Pada perkembangan teknologi, *software* memiliki peranan penting, salah satunya sebagai media informasi dan komunikasi manusia. *Hardware* yang canggih dan terbaru pun tanpa adanya *software* di dalamnya maka *hardware* tersebut tidak akan bisa digunakan.

Apabila mengetikkan kata kunci *software* yang dicari pada mesin pencarian maka akan banyak sekali situs-situs yang menyediakan *software* tersebut. Akan tetapi, terdapat kekhawatiran tentang sumber dari *software* tersebut karena banyaknya *malware/virus* yang disisipkan pada *software* yang akan diunduh. Dengan demikian perlu dibuat aplikasi yang menyediakan *software* dari sumber yang terpercaya untuk perangkat komputer pengguna.

Sistem ini dibuat dengan tujuan mempermudah pengguna untuk mendapatkan *software* yang akan di *install* atau digunakan oleh pengguna. *Software* yang disediakan bersifat *freeware* untuk mengurangi maraknya pembajakan *software*. Untuk *software* sistem operasi Windows dan Macintosh sistem dapat mengecek

*software* versi yang terakhir dengan merujuk pada situs [www.filehippo.com](http://www.filehippo.com). Untuk menu pada Linux, sistem menampilkan *sources.list* dari distro Debian.

Dengan dibuatnya *search engine repository software* ini, pengguna Linux akan dimudahkan dalam pemilihan *software* dan versi *software* yang telah disediakan oleh aplikasi. *User* dapat memilih dan meng-*custom repository* mana yang akan dimasukkan ke dalam file */etc/apt/sources.list*.

## 1.2 Rumusan Masalah

Dari uraian latar belakang masalah, maka dapat dirumuskan bahwa masalah yang melatarbelakangi penelitian ini yaitu :

1. Bagaimana membangun sebuah aplikasi *search engine repository software desktop* menggunakan teknik *web extraction*?
2. Bagaimana membangun sebuah *repository software* yang dapat selalu memperbaharui *software* secara otomatis?
3. Bagaimana mendesain beberapa tampilan utama aplikasi *search engine repository software desktop* dengan mengelompokkan berdasarkan 3 kelompok jenis *software* seperti kelompok berdasarkan sistem operasi, kelompok berdasarkan kategori, dan kelompok berdasarkan *software* yang paling baru dirilis.
4. Bagaimana melakukan *generate sources.list* untuk sistem operasi Linux Debian.

### 1.3 Batasan Masalah

1. Aplikasi *Search Engine Repository Software Desktop* berbasis *web*.
2. Aplikasi *Search Engine Repository Software Desktop* dibuat dengan menggunakan teknik *web extraction*.
3. Aplikasi *Search Engine Repository Software Desktop* menyediakan *repository* untuk *platform* sistem operasi Windows, Linux dan Macintosh.
4. Aplikasi *Search Engine Repository Software Desktop* hanya bersumber pada *source* [www.filehippo.com](http://www.filehippo.com) dan [debgen.simplylinux.ch](http://debgen.simplylinux.ch)
5. Aplikasi *Search Engine Repository Software Desktop* pada *platform* sistem operasi Linux hanya menyediakan *sources.list* aplikasi untuk distro Debian.
6. Aplikasi *Search Engine Repository Software Desktop* dapat berjalan baik di *browser* Mozilla Firefox, Google Chrome, dan Chromium.
7. Aplikasi *Search Engine Repository Software Desktop* dijalankan menggunakan *server* Debian 7.
8. Aplikasi *Search Engine Repository Software Desktop* tidak memiliki *database*.

### 1.4 Tujuan

1. Menyediakan berbagai *file* yang dibutuhkan oleh *user*, terutama *file* instalasi program/aplikasi.
2. Aplikasi *Search Engine Repository Software Desktop* dapat melakukan *generate* sumber *sources.list* aplikasi pada sistem operasi Debian 7.
3. Menambah pengetahuan secara teoritis dan praktik khusus mengenai *repository* pada Debian 7 pada sistem operasi GNU/Linux.

4. Menambah pengetahuan secara teoritis dan praktik khususnya penggunaan sistem operasi Linux.

## 1.5 Manfaat

Bagi penulis:

1. Memberikan pengetahuan tentang *web extraction*.
2. Memberikan pengalaman dalam membangun sebuah *software berbasis web*.
3. Menerapkan pengetahuan tentang *human computer interaction* pada desain tampilan antarmuka.

Bagi Pengguna:

1. Aplikasi *Search Engine Repository Software Desktop* dapat mempermudah pengguna mencari aplikasi yang dibutuhkan.
2. Mempermudah dalam mendapatkan *sources.list repository* distro Debian.

## **BAB II TINJAUAN PUSTAKA**

Penelitian mengenai *web scraping* atau *ekstraksi web* yang serupa pernah dilakukan sebelumnya dengan judul “Implementasi Sistem Sitasi Jurnal Elektronik Indonesia Berbasis Teknik Ekstraksi *Web*”. Pada penelitian tersebut peneliti membuat suatu sistem yang dapat membantu dalam proses pencarian keterkaitan kepustakaan (referensi) antara satu dokumen hasil karya penelitian dengan dokumen lainnya yang terdapat di internet.

### **2.1 *Web Scraping***

*Web Scraping* adalah proses pengambilan sebuah dokumen semi-terstruktur dari internet, umumnya berupa halaman-halaman *web* dalam bahasa *markup* seperti HTML atau XHTML. *Web scraping* menganalisis dokumen tersebut untuk diambil data tertentu dari halaman tersebut untuk digunakan bagi kepentingan lain.

*Web scraping* sering dikenal sebagai *screen scraping*. *Web Scraping* tidak dapat dimasukkan dalam bidang data mining karena data mining menyiratkan upaya untuk memahami pola semantik atau tren dari sejumlah besar data yang telah diperoleh. Aplikasi *web scraping* (juga disebut *intelligent, automated*, atau



*autonomous agents*) hanya fokus pada cara memperoleh data melalui pengambilan dan ekstraksi data dengan ukuran data yang bervariasi. Manfaat dari *web scraping* ialah agar informasi yang dikeruk lebih terfokus sehingga memudahkan dalam melakukan pencarian sesuatu. Adapun cara mengembangkan teknik *web scraping* dengan beberapa cara yaitu:

1. *Create Scraping Template*

Pembuat program mempelajari dokumen HTML dari *website* yang akan diambil informasinya untuk di tag HTML yang bertujuan untuk mengapit informasi yang diambil.

2. *Explore Site Navigation*

Pembuat program mempelajari teknik navigasi pada *website* yang akan diambil informasinya untuk ditirukan pada aplikasi *web scraping* yang akan dibuat.

3. *Automate Navigation and Extraction*

Langkah selanjutnya aplikasi *web scraping* akan mengotomatisasi informasi yang didapat dari *website* yang telah ditentukan.

4. *Extracted Data and Package History*

Informasi yang didapat tersebut akan disimpan ke dalam tabel basis data (Turland, 2010).

## 2.2 Search Engine

*Search engine* merupakan mesin pencari yang berupa sebuah *website* untuk mencari informasi yang tersimpan di dalam situs yang lain. Tiga tugas dasar sebuah *search engine* dalam cara kerjanya yaitu :

- a. Mencari di internet atau memilih bagian-bagian dari internet menurut kata-kata penting atau kunci.
- b. Memberi indeks pada kata-kata yang dicari, dan dimana mereka menemukannya.
- c. Mengizinkan pengguna untuk mencari kata-kata atau kombinasi kata yang ditemukan pada indeks (Juliasari dan Sitompul, 2012).

## 2.3 Pyramid Framework

Pyramid merupakan *open source* yang sangat umum dipakai dalam Python *web framework*. Sebagai sebuah *framework* tugas utamanya adalah membuat aplikasi *web*.

Berikut ini adalah kelebihan dari Pyramid:

1. Pyramid mencoba untuk mencoba sebagai *framework* yang menganut konsep "hanya membayar untuk apa yang kalian makan" yang akan memberikan hasil walaupun *developer* memiliki pengetahuan yang parsial. Pyramid tidak memaksa *developer web* untuk menggunakan teknologi tertentu untuk menghasilkan aplikasi.

2. Menyenangkan

Untuk mengembangkan aplikasi Pyramid tidak perlu untuk merasa takut untuk memulai sesuatu.

3. Minimalis

Pyramid menyediakan kebutuhan yang dasar seperti URL untuk pemetaan kode, *template*, keamanan, dan aset.

4. Dokumentasi

Karena Pyramid minimalis, maka relatif mudah untuk menjaga dokumentasi tetap *ter-up to date*.

5. Kecepatan

Pyramid lebih cepat daripada kebanyakan *framework Python web* lainnya untuk tugas umum seperti *template* dan *generate* respon yang sederhana.

6. Familiar

Pyramid *framework* merupakan yang pengembangan dari cara kerja otak manusia.

7. *Trustability*

Pyramid dikembangkan secara konservatif dan diuji secara mendalam.

8. Keterbukaan

Seperti Python, Pyramid *software* didistribusikan dibawah lisensi *open source* yang permisif (McDonough, 2011).

## 2.4 DNS (*Domain Name Services*)

*Server* DNS bertugas menerjemahkan IP ke nama alamat dan sebaliknya dari nama alamat ke nomor IP. DNS bersifat *client-server* sehingga administrasi cukup dilakukan di sisi *server* saja. Sedangkan pada *client* cukup dikonfigurasi 1 kali yaitu memberi cara agar mesin *client* dapat menghubungi DNS *server*.

DNS *server* terdiri dari 2 jenis *server*, yaitu:

1. *Primary Name Server* (PNS) merupakan DNS *server* yang bertanggung jawab atas resolusi domain dan subdomain yang dikelolanya.
2. *Secondary Name Server* (SNS) merupakan DNS *server* yang secara hirarki setara dengan PNS namun data-data domain dan subdomain diperoleh dengan cara menyalin dari PNS (*Anonymous* , 2011).

## 2.5 Metode Pengembangan Aplikasi

Perancangan aplikasi memerlukan suatu metode atau langkah untuk membangun atau mengembangkan aplikasi. Metode pembuatan aplikasi yang digunakan dalam penelitian ini menggunakan metode *Extreme Programming*. *Extreme Programming* dipopulerkan oleh Beck (1999) sebagai metode atau pendekatan untuk mengembangkan sistem perangkat lunak.

Metode *Extreme Programming* dikembangkan karena pada metodologi tradisional terdapat banyak hal yang membuat proses pengembangan tidak dapat berhasil dengan baik sesuai dengan perubahan kebutuhan yang cepat dan tuntutan dari *user*. Beberapa hal yang menjadi dasar dalam tahapan proses pengembangan perangkat lunak dijabarkan sebagai berikut:

1. *Communication* (Komunikasi)

*Communication* (komunikasi) merupakan hal yang penting dalam sebuah *teamwork*, bagaimana individu dapat berkomunikasi dengan baik antara satu dengan yang lain untuk men-*development software*.

2. *Courage* (Keberanian)

*Teamwork* dan *software developer* harus memiliki keyakinan dan integritas terutama saat menghadapi tekanan dari *client*.

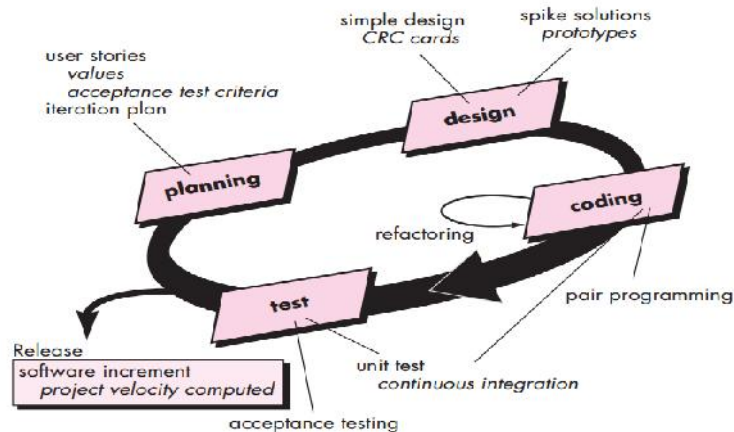
3. *Simplicity* (Kesederhanaan)

Maksud aspek *simplicity* adalah mengerjakan dengan cara sederhana seperti menggunakan metode yang pendek dan sederhana, desain yang tidak terlalu rumit, *unused features* dihilangkan, dan berbagai proses penyederhanaan aspek lainnya.

4. *Feedback* (Umpan Balik)

Setiap permasalahan dan perubahan yang terjadi diungkapkan dan anggota tim diberi kesempatan untuk mengutarakan pendapat masing-masing (Beck, 1999).

*Extreme Programming* menggunakan suatu pendekatan berorientasi objek sebagai paradigma pengembangan yang disukai dan mencakup seperangkat aturan dan praktik-praktik yang terjadi dalam konteks yang terdiri dari 4 *framework* aktivitas yaitu: *planning* (perencanaan), *design* (desain), *coding* (pengkodean), dan *testing* (pengujian). Gambar 2.1 berikut ini adalah ilustrasi dari proses *Extreme Programming* yang berkaitan dengan aktivitas kerangka kerja.



Gambar 2.1 Proses *Extreme Programming*

### 1. *Planning* (Perencanaan)

Aktivitas perencanaan dimulai dengan mendengarkan/melihat sekumpulan kebutuhan aktivitas yang memungkinkan anggota teknis dari tim *Extreme Programming* untuk memahami konteks bisnis untuk *software* yang akan dikembangkan.

### 2. *Design* (Desain)

*Extreme Programming* di desain dengan mengikuti prinsip KIS (*keep it simple*). Sebuah desain sederhana selalu lebih disukai daripada representasi yang kompleks. Sebuah gagasan utama dalam *Extreme Programming* adalah desain terjadi baik sebelum dan sesudah *coding* dimulai. *Refactoring* berarti bahwa desain terjadi terus menerus selama sistem dibangun.

### 3. *Coding* (Pengkodean)

Setelah menyelesaikan serangkaian desain awal, pengembang tidak langsung melakukan *coding*, melainkan mengembangkan serangkaian tes terlebih dahulu. setelah itu baru pengembang melakukan *coding*. Setelah *coding* selesai dibuat,

maka dapat diuji untuk mendapatkan *feedback* dari *user*. Pada metode *Extreme Programming*, konsep utama selama melakukan *coding* adalah *pair programming*. *Extreme Programming* merekomendasikan bahwa dua orang yang bekerja bersama-sama pada satu *project* akan memberikan hasil yang lebih baik. sebagai *pair programmer*, *coding* yang dikembangkan dilakukan terintegrasi dengan pekerjaan orang lain.

#### 4. *Testing* (Pengujian)

Unit tes yang dibuat harus dilaksanakan menggunakan kerangka kerja yang memungkinkan pengujian dapat dilakukan dengan mudah dan berulang-ulang. *Testing* pada *Extreme Programming* disebut juga tes pengguna, dimana pengguna fokus pada fitur sistem secara keseluruhan dan fungsi yang terlihat oleh pengguna (Pressman, 2010).

### **2.6 Black Box Testing**

*Black box testing* merupakan suatu teknik pengujian perangkat lunak dengan berfokus pada persyaratan fungsional. *Black box testing* memungkinkan perancang perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. *Black box testing* berusaha menemukan kesalahan dalam kategori sebagai berikut:

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses *database* eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.

*Black box testing* diaplikasikan selama tahap akhir pengujian, karena *black box* memperhatikan struktur kontrol, maka perhatian berfokus pada domain informasi (Pressman, 2010).



## **BAB III METODE PENELITIAN**

### **3.1 Tempat dan Waktu Penelitian**

Penelitian ini dilakukan di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung dan di Linux Lampung yang beralamatkan di Jalan H. Abdul Mutolib no. 15 Gedong Air, Bandar Lampung. Penelitian ini dilakukan pada semester ganjil tahun ajaran 2014/2015.

### **3.2 Alat Pendukung**

Analisis kebutuhan merupakan analisis terhadap komponen-komponen yang digunakan untuk pembuatan sistem. Dalam hal ini, komponen yang dibutuhkan terbagi menjadi 2 macam, yaitu komponen perangkat keras dan perangkat lunak.

#### **3.2.1 *Hardware***

Perangkat keras adalah alat yang digunakan untuk menunjang dalam pembuatan sistem. Dalam pembuatan sistem ini, perangkat keras yang digunakan yaitu laptop dengan spesifikasi sebagai berikut :

1. *Notebook* DELL Inspiron N4050 dengan CPU *core* i3
2. RAM 4GB DDR3
3. HDD 500 GB

### 3.2.2 *Software*

Perangkat lunak adalah program atau aplikasi komputer lain yang dibutuhkan untuk membangun sistem. Perangkat lunak digunakan untuk mengelola seluruh sumber daya atau perangkat keras.

#### 3.2.2.1 Sistem Operasi

Dalam perancangan sistem ini, sistem operasi yang digunakan adalah sistem operasi GNU/Linux dengan distribusi Debian versi 7, karena sistem operasi GNU/Linux telah menyertakan *tools* atau aplikasi untuk keperluan perancangan jaringan, serta Linux yang bersifat *open source* sehingga tidak perlu membayar lisensi dan meminimalis penggunaan perangkat lunak ilegal.

#### 3.2.2.2 *Software Aplikasi*

*Software* aplikasi adalah suatu sub kelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan, *software* aplikasi yang dibutuhkan dalam membangun sistem ini adalah *software* yang mendukung kebutuhan jaringan, antara lain adalah :

- a. *Bluefish Editor*, merupakan *software* untuk membuat *file* HTML, PHP dan CSS.
- b. Python versi 3.4.1 sebagai bahasa pemrograman yang digunakan untuk membuat sistem.
- c. Pyramid yang digunakan sebagai *framework* dalam membangun *website*

- d. AngularJS, sama seperti Pyramid yang berfungsi sebagai *framework* dalam membangun *website*.

### **3.3 Metode Pengembangan Sistem**

Pada penelitian ini sistem yang dibuat menggunakan metode *Extreme Programming*. Metode *Extreme Programming* ini merupakan metode yang sangat fleksibel dan cocok dipakai untuk mengembangkan *software* dalam waktu singkat. Dalam metode ini terdapat empat tahapan dalam pengembangan sistem yaitu tahap *planning* (perencanaan), tahap desain, tahap *coding* dan tahap *testing*.

#### **3.3.1 Planning (Perencanaan)**

Pada tahap ini pengembang bertemu dengan *client* untuk membahas aplikasi *Search Engine Repository Software Desktop* yang akan dibuat. Pengembang melakukan komunikasi dengan pihak *client* untuk mendapatkan gambaran aplikasi yang diinginkan oleh pihak *client*. Pada tahap ini dilakukan komunikasi secara intensif hingga pengembang dan *client* mendapatkan gambaran yang sama tentang aplikasi yang akan dibuat.

#### **3.3.2 Design (Desain)**

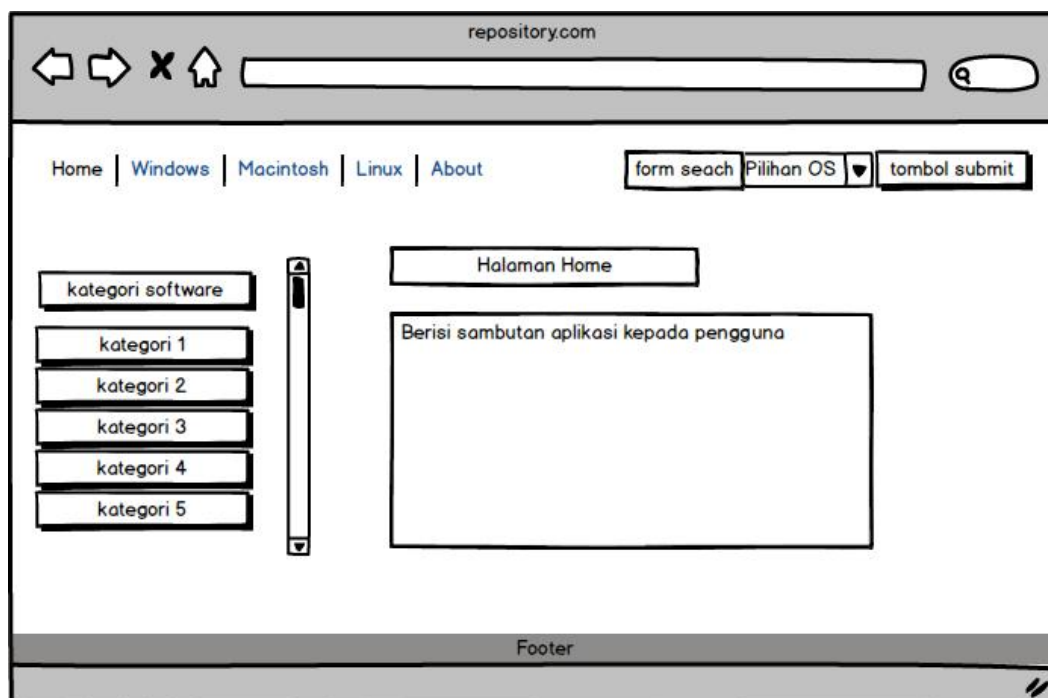
Setelah tahap *planning* selesai dilakukan maka pengembang membuat desain tentang aplikasi *Search Engine Repository Software Desktop*. Desain ini nantinya akan dikomunikasikan kepada pihak *client* untuk melihat apakah desain yang dibuat sudah sesuai dengan keinginan pihak *client*. Pada pembuatan sistem ini menggunakan *mockup* dan *sequence diagram* sebagai desainnya.

### 3.3.2.1 Mockup

*Mockup* adalah sketsa atau rancangan awal dari sebuah desain *web* yang dibuat menggunakan *software* pengedit gambar atau pembuat *mockup*. *Mockup* berfungsi sebagai acuan kerja pembuatan *website* agar tidak menyimpang dari tujuan awal membuatnya. Terdapat beberapa desain halaman aplikasi *Search Engine Repository Software Desktop* yang dibuat. Desain tersebut meliputi desain halaman *home*, desain halaman Windows, desain halaman Macintosh, desain halaman Linux, dan desain halaman *generate sources.list*.

#### 1. Mockup Halaman Home

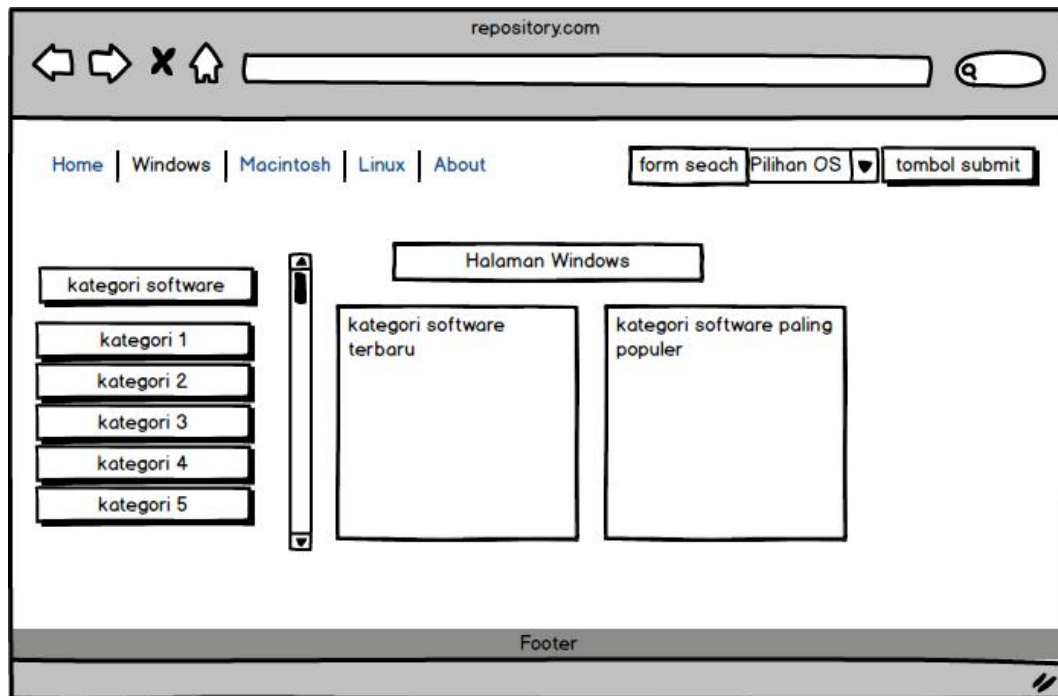
Pada *mockup* halaman *home*, apabila pengguna meng-*klik* tombol *home* maka sistem akan menampilkan sambutan sistem kepada pengguna. Desain menu *home* dapat dilihat pada Gambar di bawah ini.



Gambar 3.1 Mockup Halaman Home

## 2. *Mockup* Halaman Windows

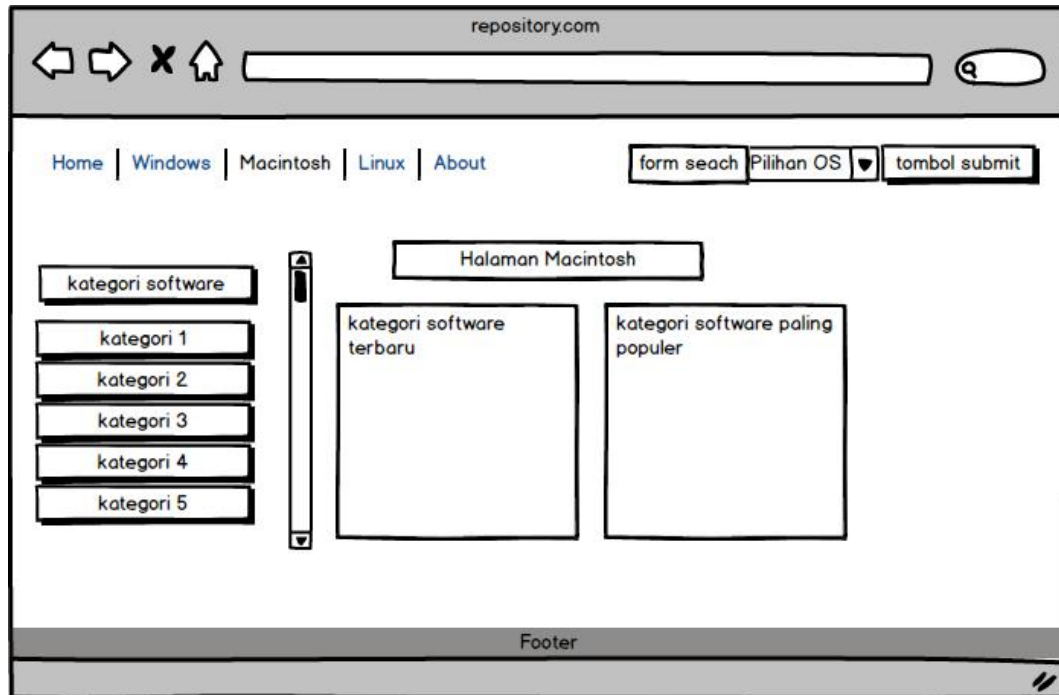
Pada *mockup* halaman Windows, sistem dapat menampilkan kategori *software* terbaru, kategori *software* paling populer, dan menampilkan *software* berdasarkan kategori. Adapun gambaran desain dapat dilihat pada Gambar 3.2 berikut ini.



Gambar 3.2 *Mockup* Halaman Windows

## 3. *Mockup* Halaman Macintosh

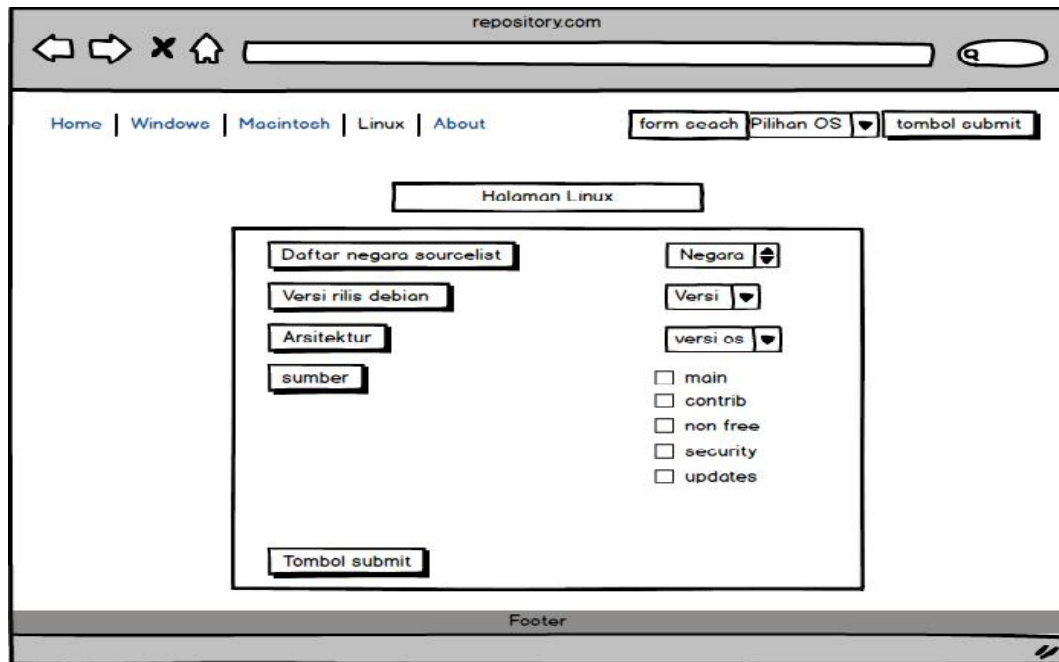
Pada *mockup* halaman Macintosh, sistem dapat menampilkan kategori *software* terbaru, kategori *software* paling populer, dan menampilkan *software* berdasarkan kategori. Adapun gambaran desain dapat dilihat pada Gambar 3.3 berikut ini.



Gambar 3.3 *Mockup* Halaman *Macintosh*

#### 4. *Mockup* Halaman *Linux*

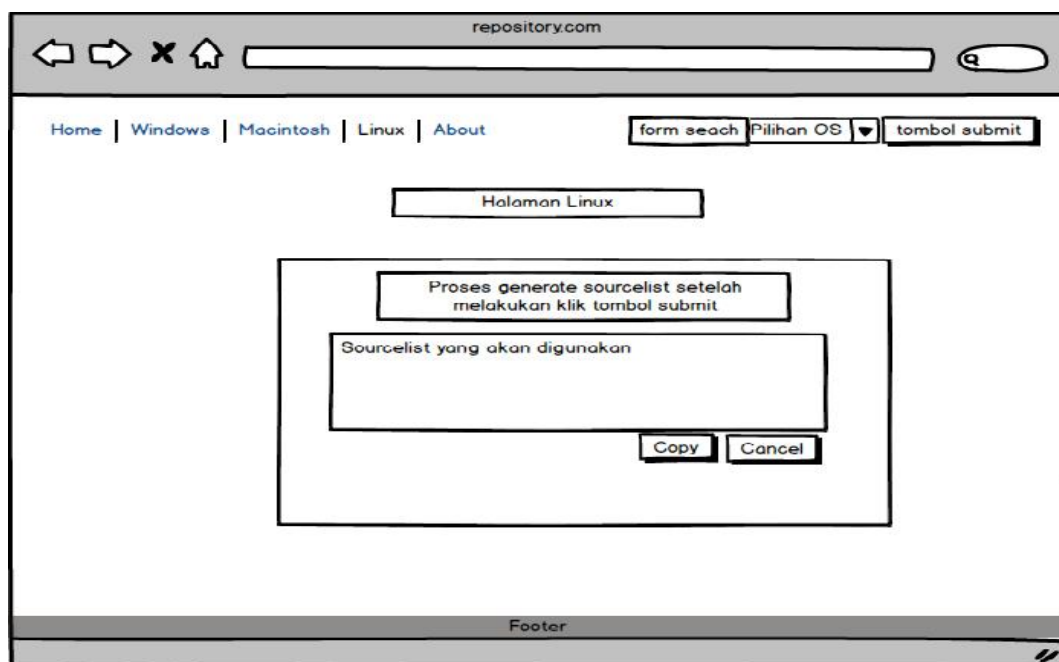
Pada *mockup* halaman *Linux*, sistem dapat menampilkan detail *sources.list* seperti asal negara, versi *Debian* yang akan digunakan, arsitektur yang digunakan dan memilih sumber yang akan digunakan. Adapun gambaran desain dapat dilihat pada Gambar 3.4 berikut ini.



Gambar 3.4 *Mockup* Halaman Linux

### 5. *Mockup* Halaman *Generate Sources.list*

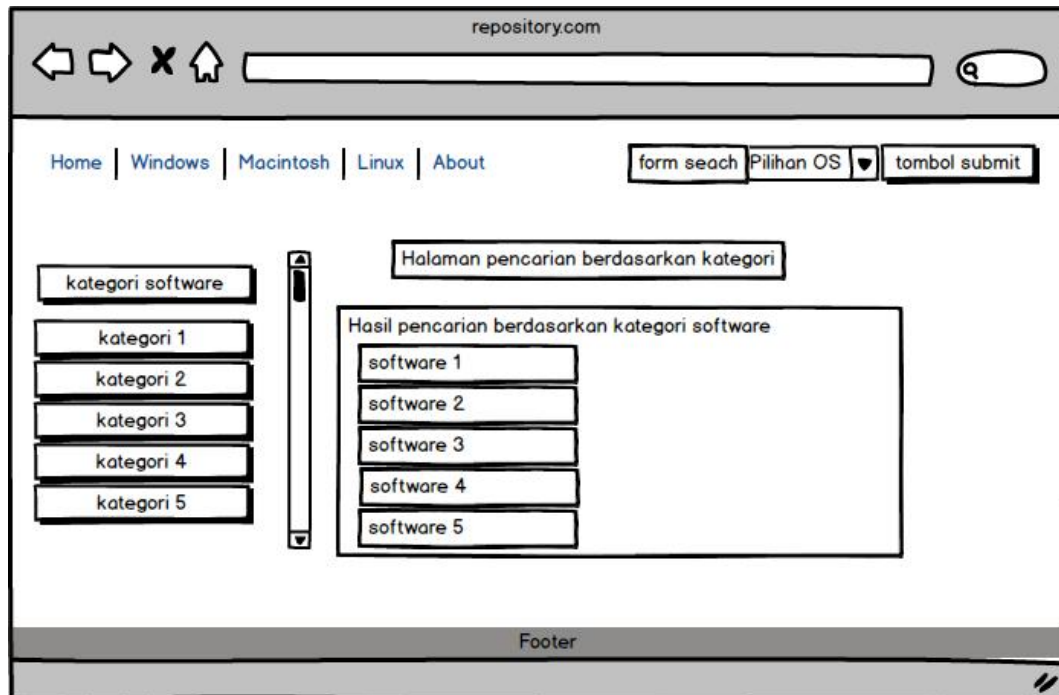
Pada *mockup* halaman ini, sistem dapat menampilkan hasil dari proses *generate sources.list* yang telah dipilih.



Gambar 3.5 *Mockup* Halaman *Generate Sources.list*

## 6. *Mockup* Halaman Pencarian Berdasarkan Kategori

Pada *mockup* halaman pencarian berdasarkan kategori, sistem dapat menampilkan daftar *software* berdasarkan jenis kategori *software* tersebut.

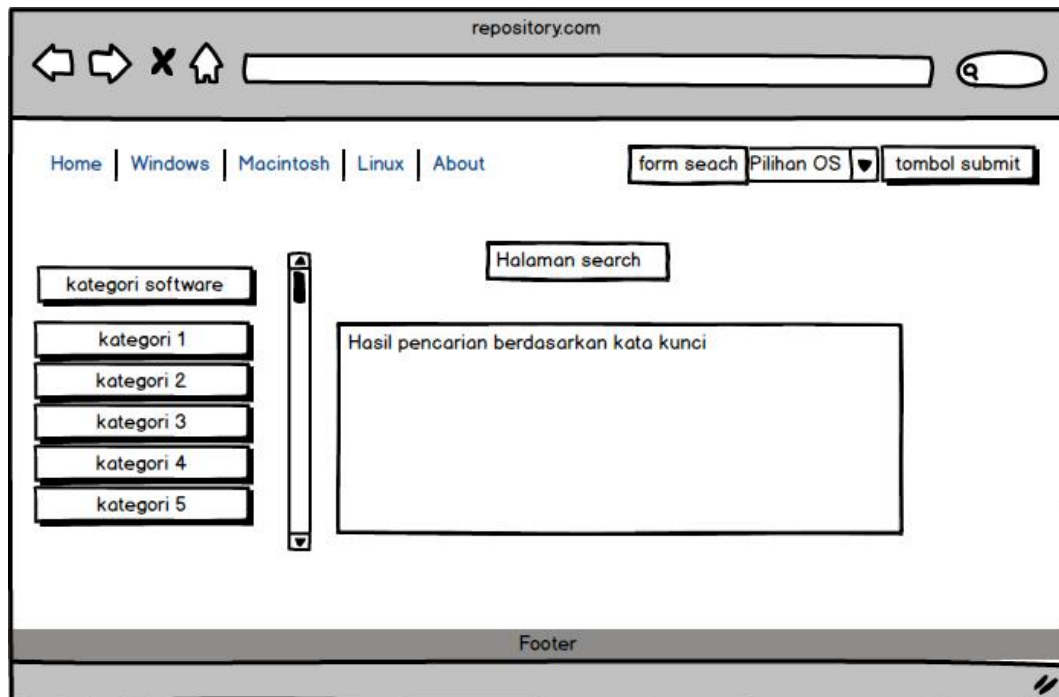


Gambar 3.6 *Mockup* Halaman Pencarian Berdasarkan Kategori

## 7. *Mockup* Halaman Pencarian Berdasarkan Kata Kunci

Pada *mockup* halaman pencarian berdasarkan kata kunci, sistem dapat menampilkan daftar *software* berdasarkan kata kunci yang diketikkan oleh *user* pada *form search*.

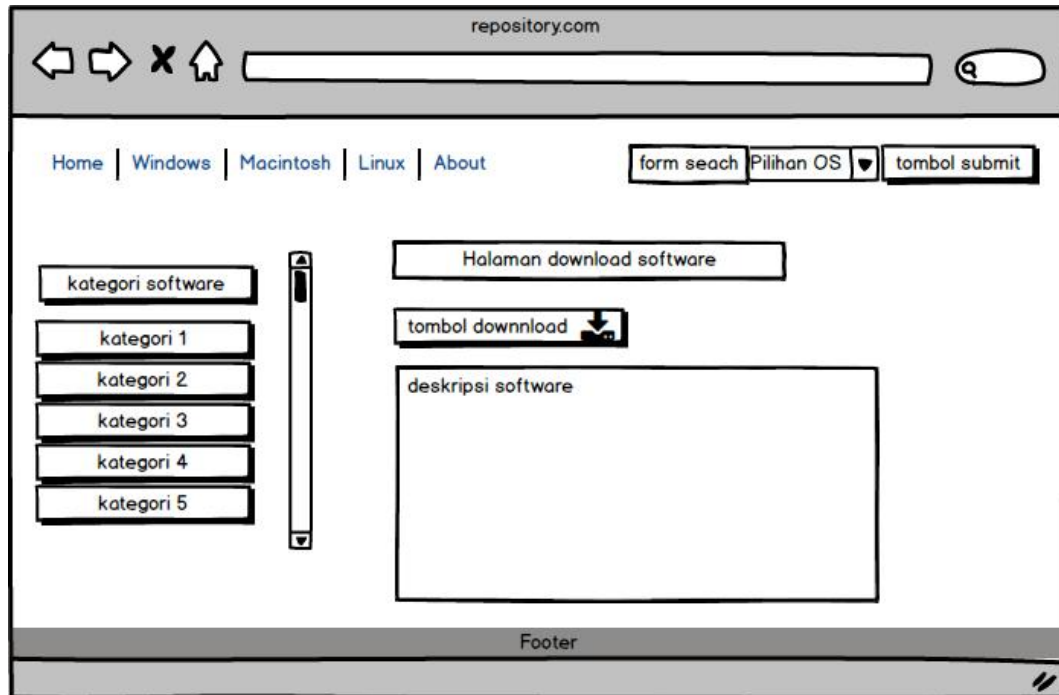




Gambar 3.7 *Mockup* Pencarian Berdasarkan Kata Kunci

## 8. *Mockup* Halaman *Download*

Pada *mockup* halaman *download*, sistem dapat menampilkan tombol *download* yang berfungsi untuk men-*download software* beserta deskripsi dari *software* yang akan di-*download* oleh *user*.



Gambar 3.8 *Mockup Halaman Download*

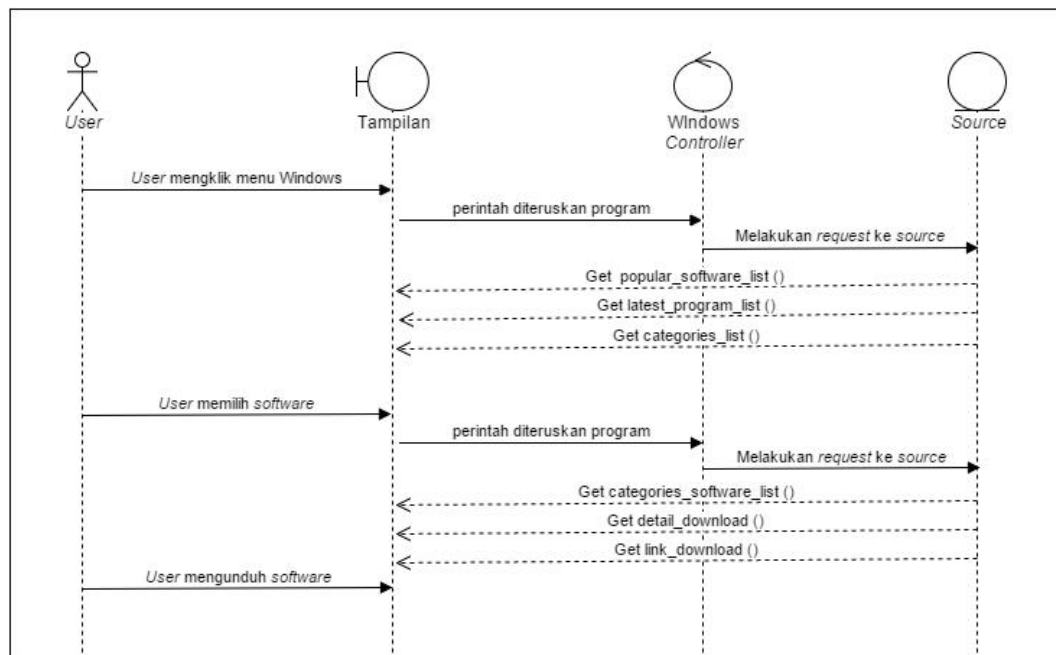
### 3.3.2.2 *Desain Sequence Diagram*

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

*Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. Diagram ini diawali dari apa yang *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

## 1. *Sequence Diagram Halaman Windows*

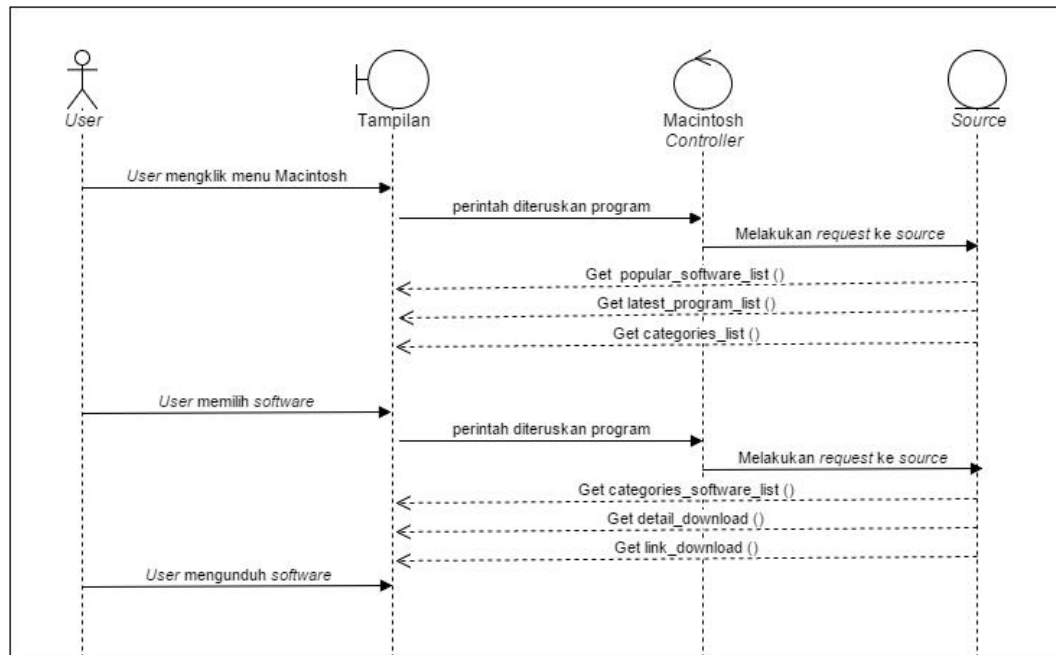
Pada Gambar 3.9 *user* dapat memilih *software* Windows berdasarkan kategori *software*, *popular software*, dan *latest software*. Apabila *user* telah memilih *software* yang diinginkan maka sistem dapat menampilkan detail *software* tersebut lalu menampilkan *link download* agar *user* dapat men-*download file* tersebut.



Gambar 3.9 *Sequence Diagram Halaman Windows*

## 2. *Sequence Diagram Halaman Macintosh*

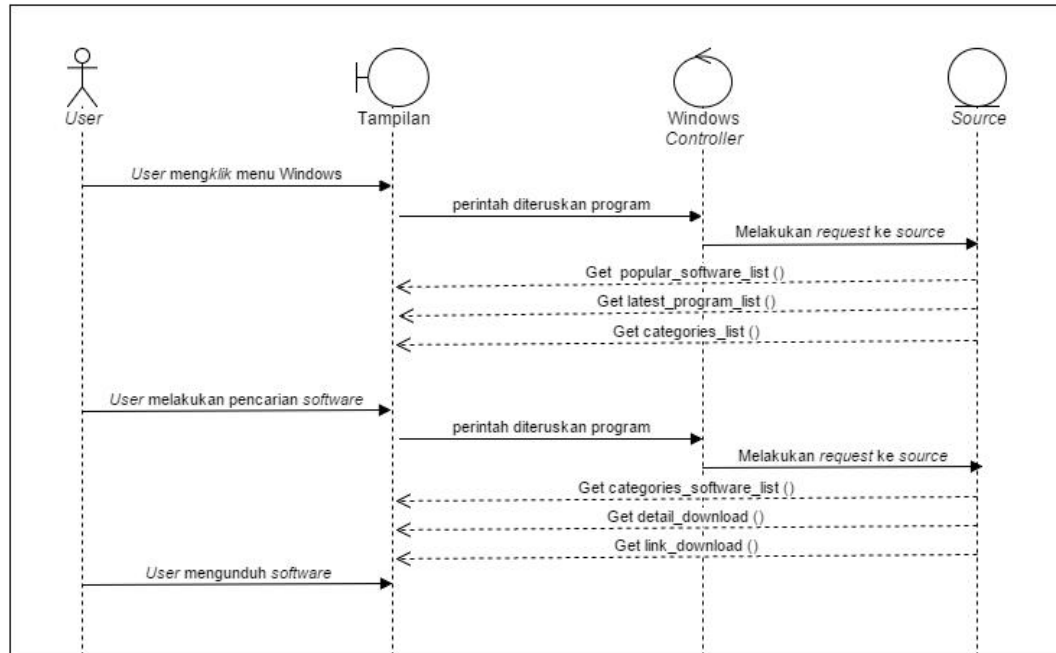
Pada Gambar 3.910 *user* dapat memilih *software* Macintosh berdasarkan kategori *software* *popular software*, dan *latest software*. Apabila *user* telah memilih *software* yang diinginkan maka sistem dapat menampilkan detail *software* tersebut lalu menampilkan *link download* agar *user* dapat men-*download file* tersebut.



Gambar 3.10 *Sequence Diagram* Halaman Macintosh

### 3. *Sequence Diagram* Pencarian Software Windows

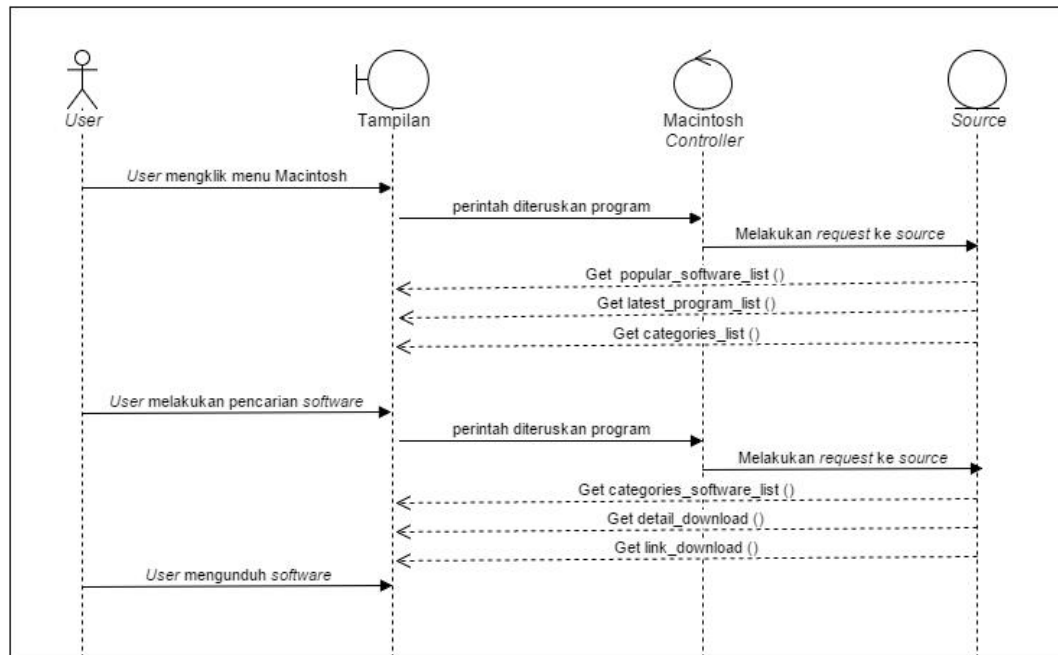
Pada Gambar 3.11 *user* dapat mencari *software* Windows berdasarkan kategori *software*, *popular software*, dan *latest software*. *User* juga dapat melakukan pencarian dengan mengetikkan *keyword* pada *form* yang telah disediakan. Apabila *user* telah memilih *software* yang diinginkan maka sistem dapat menampilkan detail *software* tersebut lalu menampilkan *link download* agar *user* dapat *download file* tersebut.



Gambar 3.11 *Sequence Diagram* Pencarian *Software Windows*

#### 4. *Sequence Diagram* Pencarian *Software Macintosh*

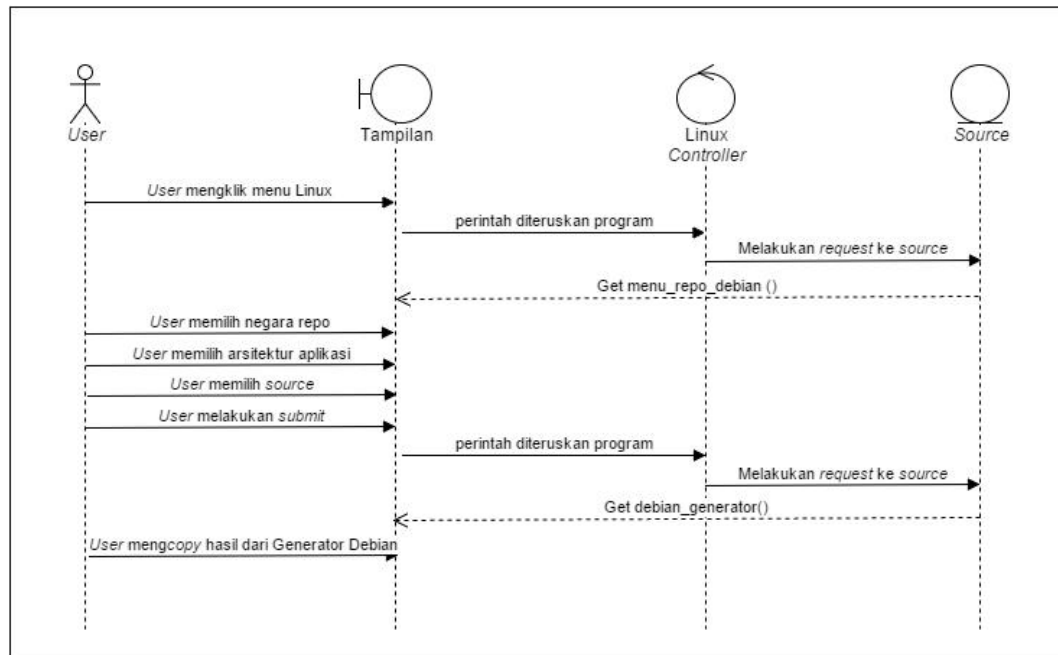
Pada Gambar 3.11 *user* dapat mencari *software Macintosh* berdasarkan kategori *software*, *popular software*, dan *latest software*. *User* juga dapat melakukan pencarian dengan mengetikkan *keyword* pada *form* yang telah disediakan. Apabila *user* telah memilih *software* yang diinginkan maka sistem dapat menampilkan detail *software* tersebut lalu menampilkan *link download* agar *user* dapat *download file* tersebut



Gambar 3.12 *Sequence Diagram* Pencarian *Software* Macintosh

## 5. *Sequence Diagram* Debian Generator

Pada Gambar 3.13 *user* meng-klik menu Linux, kemudian sistem akan menampilkan negara asal repo, arsitektur aplikasi, dan *source* yang akan digunakan. *User* dapat memilih sesuai dengan *software* yang dibutuhkan. Selanjutnya apabila *user* telah memilih *software* yang diinginkan maka sistem akan melakukan *generate sources.list* dan *user* dapat meng-copy data tersebut ke dalam file */etc/apt/sources.list*.



Gambar 3.13 *Sequence Diagram Debian Generator*

### 3.3.3 Coding (Pengkodean)

Setelah menyelesaikan serangkaian desain awal, pengembang dapat melakukan *coding*. Setelah *coding* selesai dibuat, maka dapat diuji untuk mendapatkan *feedback* dari *user*.

### 3.3.4 Testing (Pengujian Sistem)

Apabila seluruh kebutuhan aplikasi telah terpasang di perangkat komputer yang digunakan dan *coding* telah selesai dilakukan, maka selanjutnya dilakukan pengujian terhadap sistem. Pengujian dilakukan dengan menggunakan metode *blackbox*. Pengujian dilakukan berdasarkan apakah kebutuhan fungsional sistem sudah berjalan dengan baik. Pengecekan dilakukan dengan cara menguji apakah sistem dapat mengunduh *file* yang tersedia di halaman menu. Pengecekan juga

dilakukan apakah daftar *software* yang ada pada sistem merupakan versi terbaru. Sedangkan pada menu Debian dilakukan pengujian apakah sistem dapat menampilkan/melakukan *generate file sources.list* yang telah dibuat oleh *user*.



## **BAB V**

### **SIMPULAN DAN SARAN**

#### **5.1 Simpulan**

Pada penelitian ini dapat ditarik kesimpulan sebagai berikut:

1. Aplikasi *Search Engine Repository Software Desktop* sangat bergantung pada tampilan *web* yang dijadikan sumber data. Perubahan *interface* pada *web* yang dijadikan sumber data akan mengakibatkan kegagalan pada pengambilan data, sehingga perlu dilakukan pengambilan data kembali.
2. Aplikasi yang dibuat menyediakan *software* yang bersifat *freeware* untuk sistem operasi Windows dan Macintosh.
3. Aplikasi yang dibuat menyediakan *repository* untuk sistem operasi Linux dengan distro Debian.

#### **5.2 Saran**

Adapun saran pada sistem yang telah dibuat adalah agar menggunakan koneksi internet yang stabil dan cepat, karena sistem membutuhkan kecepatan dalam transmisi data dan *request* yang cepat dari *server*.

## DAFTAR PUSTAKA

- Anonymous. 2011. *Administrasi Jaringan dengan Linux Ubuntu 11*. Andi. Yogyakarta.
- Beck, Kent. 1999. *Extreme Programming Explained*. Addison-Wesley. United States.
- Juliasari, Noni dan Joseph Sitompul. 2012. Aplikasi *Search Engine* dengan Metode *Depth First Search (DFS)*. BIT. Vol 9 (1).
- McDonough, Chris. 2011. *The Pyramid Web Application Development Framework*. Agendaless Consulting. United States.
- Pressman, Roger. 2010. *Software Engineering: A Practitioner's Approach, Seventh Edition*. McGraw-Hill. New York.
- Turland, Matthew. 2010. *php/architect's Guide to Web Scraping with PHP*. Marco Tabini. Canada.