

**TUGAS AKHIR**

***AUTOMATIC UPDATE WEB FILTERING BERBASIS SQUID  
DENGAN BANTUAN SEARCH ENGINE***

**Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana  
Strata-1 Teknik Informatika**



**Oleh :**

**Nama : Citra Arfanudin**

**Nim : 07523261**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA  
2012**

**LEMBAR PENGESAHAN DOSEN PEMBIMBING**

***AUTOMATIC UPDATE WEB FILTERING BERBASIS SQUID  
DENGAN BANTUAN SEARCH ENGINE***

**TUGAS AKHIR**



Yogyakarta, Mei 2012

**Dosen pembimbing**

**(Syarif Hidayat ,S.Kom., M.I.T.)**

LEMBAR PENGESAHAN DOSEN PENGUJI

*AUTOMATIC UPDATE WEB FILTERING BERBASIS SQUID DENGAN BANTUAN SEARCH ENGINE*

TUGAS AKHIR

Oleh :

Nama : CITRA ARFANUDIN

No.Mahasiswa : 07 523 261

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta,  
Tim Penguji,

Syarif Hidayat ,S.Kom., M.I.T.  
Ketua

Raden Teduh Dirgahayu , Dr.S.T., M.Sc.  
Anggota I

Hendrik ,S.T., M.Eng.  
Anggota II

Mengetahui,

Ketua Jurusan Teknik Informatika  
Fakultas Teknologi Industri  
Universitas Islam Indonesia



Yudi Prayudi, S.Si., M.Kom.

## **LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR**

Saya yang bertandatangan di bawah ini :

**Nama : CITRA ARFANUDIN**

**No. Mahasiswa : 07 523 261**

Menyatakan bahwa seluruh komponen dan isi dalam Laporan Tugas Akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti bahwa ada beberapa bagian dari karya ini adalah bukan hasil karya saya sendiri, maka saya siap menanggung risiko dan konsekuensi apapun.

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, Mei 2012

CITRA ARFANUDIN

## **HALAMAN PERSEMBAHAN**

Kupersembahkan Tugas Akhir Ini Kepada:

Allah SWT yang telah mendengarkan semua doaku serta selalu melimpahkan rahmat dan hidayah-Nya kepadaku

Bapak dan Ibuku atas semua doa, kasih sayang, dukungan dan didikannya kepadaku

Keluargaku, kakak-kakakku

(Mba Risa, Mas Shofi, Mas Wanto. Yu Heni, Yu Heti, Mba Sri, Mba Fitri )  
Dan Keponakanku Tersayang (Daniyal, Jazlyn, Azalea, Zinnia dan Caesa)

## **HALAMAN MOTTO**

“tidak ada masalah yang tidak dapat diselesaikan, jika kita menyerah, habislah kita” The Billionaire

“Waktu tidak dapat menunggu, kerjakanlah apa yang kamu bisa kerjakan sekarang”

“Aku adalah apa yang aku pikirkan”

## KATA PENGANTAR



Alhamdulillah Rabbil'alamin. Puji dan syukur kehadiran Allah SWT yang telah melimpahkan rahmat, nikmat dan hidayah-Nya. Shalawat serta salam senantiasa tercurah kepada junjungan kita Rasulullah Muhammad SAW beserta para keluarga, sahabat serta para pengikutnya, sehingga terselesaikannya tugas akhir dengan judul “*AUTOMATIC UPDATE WEB FILTERING BERBASIS SQUID DENGAN BANTUAN SEARCH ENGINE*”.

Laporan tugas akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar sarjana pada Jurusan Teknik Informatika Fakultas Teknologi Industri, Universitas Islam Indonesia.

Penulisan dan penyelesaian tugas akhir ini tidak lepas dari saran, bimbingan, dukungan serta bantuan dari berbagai pihak. Untuk itu pada kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Allah SWT atas segala berkah dan rahmat-Nya sehingga Tugas Akhir ini dapat diselesaikan.
2. Yang tercinta ibunda Tarsinah, dan ayahanda Kurdi MS yang telah mendidikku sehingga menjadi pribadi yang seperti ini.
3. Saudara dan Sahabat-sahabat atas kasih sayang, segala limpahan do'a dan dukungan.
4. Bapak Yudi Prayudi S.Si,M.Kom selaku Ketua Jurusan Teknik Informatika FTI UII.
5. Bapak Syarif Hidayat ,S.Kom., MIT sebagai Dosen Pembimbing dalam Penyusunan Tugas Akhir ini.
6. Seluruh Dosen di Jurusan Teknik Informatika Universitas Islam Indonesia yang dengan sabar mewariskan ilmunya kepada kami, serta memberikan motivasi dan semangat.
7. Teman-teman angkatan 2007 Informatika UII (Include).
8. All dedemit jarkom yang sudah menemani disaat stress dan sepi datang.

9. All G10 Members, thanks 4 this 5 years.
10. Spesial for Timur sahadewa. Terimakasih atas ilmu dan bantuannya.
11. Semua pihak yang tidak bisa disebutkan satu per satu.

Saya berharap laporan ini dapat menjadi sesuatu yang bermanfaat bagi kita semua untuk menambah wawasan dan pengetahuan, Amin.

Wassalamu'alaikum Wr.Wb

Yogyakarta, Mei 2012

Citra Arfanudin



## INTISARI

Squid adalah perangkat lunak *Proxy Server* yang bekerja pada Sistem Operasi Linux dan yang memiliki berbagai kegunaan, dari mempercepat *web Server* dengan meng-*cache* permintaan yang diulang-ulang dan juga meng-*cache web*, selain itu berfungsi untuk membantu keamanan dengan menyaring lalu lintas data. Banyak *website* yang mengancam keamanan komputer menuntut sebuah sistem yang dapat memblokir jaringan lokal agar tidak mengakses *website* yang mengancam tersebut.

*Squidsearch* merupakan *proxy squid* yang didalamnya *Administrator* dapat melakukan konfigurasi berbasis *web* sehingga dapat melakukan pengaturan dan kontrol terhadap manajemen daftar *website* yang dapat atau dilarang untuk diakses. Sehingga *Administrator* dapat dengan mudah melakukan kontrol terhadap jaringan.

Aplikasi *Squidsearch* dapat melakukan *update* basisdata *proxy* yang berisi *blocked list website* yang tidak dapat diakses dengan cara menggunakan *service crontab* dan dengan bantuan *search engine*.

Fungsi manajemen yang ada didalam *Squidsearch* adalah manajemen *urlregex*, manajemen *Keyword Search*, manajemen *blocked list*, manajemen *userpanel*. Dengan menggunakan sistem ini *Administrator* cukup untuk melakukan *update* basisdata *blocked list proxy*.

Keyword : *search engine, squid, web filtering, proxy, squid-search*

## TAKARIR

<i>cache</i>	tempat penyimpanan pada <i>proxy</i>
<i>gateway</i>	pintu menuju jaringan berbeda
<i>layer</i>	lapisan
<i>root</i>	akar
<i>redirection</i>	mengarahkan
<i>resources</i>	sumber daya
<i>router</i>	perangkat untuk menghubungkan jaringan berbeda
<i>service</i>	layanan
<i>sharing</i>	berbagi
<i>banned</i>	dilarang
<i>update</i>	memperbaharui
<i>caching</i>	penyimpanan object yang sudah di akses
<i>netizen</i>	pengguna <i>internet</i>
<i>search engine</i>	mesin pencari
<i>website</i>	situs
<i>webpage</i>	halaman <i>web</i>
<i>filtering</i>	penyaringan
<i>blocked</i>	terblokir

## DAFTAR ISI

HALAMAN JUDUL .....	i
LEMBAR PENGESAHAN DOSEN PEMBIMBING .....	ii
LEMBAR PENGESAHAN DOSEN PENGUJI .....	iii
LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN .....	v
HALAMAN MOTTO.....	vi
KATA PENGANTAR .....	vii
INTISARI .....	ix
TAKARIR .....	x
DAFTAR ISI .....	xi
DAFTAR GAMBAR.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian .....	3
1.6 Metodologi Penelitian .....	3
1.6.1 Studi Pustaka .....	3
1.6.2 Implementasi dan Konfigurasi Sistem .....	3
1.7 Sistematika Penulisan .....	4
BAB II LANDASAN TEORI .....	6
2.1 Konsep Kerja <i>Filtering Proxy</i> .....	6
2.1.1 <i>Gateway</i> .....	7
2.1.2 <i>Firewall</i> .....	7
1. <i>Packet Filtering</i> .....	8
2. <i>Domain Filtering</i> .....	9
2.2 <i>Search engine</i> .....	11

BAB III METODOLOGI .....	13
3.1 Analisis Masalah .....	13
3.2 Gambaran Umum Sistem .....	14
3.2.1 Desain dan Alur Data Proses Squidsearch .....	15
3.2.2 Cara Kerja Sistem <i>Search Engine</i> .....	16
3.3 <i>Analisis</i> dan persiapan kebutuhan sistem.....	17
3.3.1 Kebutuhan perangkat keras .....	17
3.3.2 Kebutuhan perangkat lunak.....	18
3.4 Perancangan Aplikasi <i>Squidsearch</i> .....	19
3.4.1 DFD Level 0 .....	19
3.4.2 DFD Level 1 .....	20
3.4.3 DFD Level 2 Proses 2 .....	21
3.4.4 DFD Level 2 Proses 3 .....	21
3.4.5 DFD Level 2 Proses 4 .....	22
3.4.6 DFD Level 2 Proses 5 .....	22
3.4.7 DFD Level 2 Proses 6 .....	23
3.5 Perancangan Tampilan <i>Squidsearch</i> .....	23
3.5.1 Tampilan Halaman <i>Login Squidsearch</i> .....	23
3.5.2 Tampilan Halaman Utama <i>Squidsearch</i> .....	24
3.5.3 Tampilan Halaman <i>Admin Manage Userpanel</i> .....	24
3.5.4 Tampilan Halaman <i>Manage Keyword Search Squidsearch</i> .	25
3.5.5 Tampilan Halaman <i>Add Keyword Search Squidsearch</i> .....	25
3.5.6 Tampilan Halaman <i>Manage Url Regex Squidsearch</i> .....	26
3.5.7 Tampilan Halaman <i>Add Url Regex Squidsearch</i> .....	26
3.5.8 Tampilan Halaman <i>Manage Blocked List Squidsearch</i> .....	27
3.5.9 Tampilan Halaman <i>Add Blocked List Squidsearch</i> .....	27
3.5.10 Tampilan Halaman <i>Proxy Setting Squidsearch</i> .....	28
BAB IV HASIL DAN PEMBAHASAN .....	30
4.1 Implementasi Secara Umum .....	30
4.2 Tahapan Implementasi Perangkat Lunak .....	30
4.3 Implementasi Tahap Perancangan .....	31

4.3.1	Instalasi dan konfigurasi <i>proxy</i> .....	31
4.3.2	Instalasi dan konfigurasi basisdata MySQL.....	34
4.3.3	Instalasi <i>phpmyadmin</i> .....	34
4.3.4	Instalasi dan konfigurasi <i>Web Server Apache2</i> .....	36
4.3.5	Instalasi modul <i>php5-curl</i> .....	36
4.3.6	Instalasi <i>crontab</i> .....	36
4.3.7	Pembuatan <i>shell script</i> .....	36
4.3.8	Pembuatan <i>script interface</i> untuk konfigurasi <i>Server</i> .....	38
3.4	Hasil Perancangan .....	42
3.4.1	Halaman <i>Login Squidsearch</i> .....	42
3.4.2	Halaman <i>Manage Url Regex</i> .....	42
3.4.3	Halaman <i>Add Url Regex</i> .....	43
3.4.4	Halaman <i>Edit Url Regex</i> .....	44
3.4.5	Halaman <i>Manage Keyword Search</i> .....	45
3.4.6	Halaman <i>Add Keyword Search</i> .....	46
3.4.7	Halaman <i>Edit Keyword Search</i> .....	48
3.4.8	Halaman <i>Manage Blocked List</i> .....	48
3.4.9	Halaman <i>Add Blocked List</i> .....	49
3.4.10	Halaman <i>Edit Blocked List</i> .....	50
3.4.11	Halaman <i>Proxy Setting</i> .....	51
3.4.12	Halaman <i>Managemen Userpanel</i> .....	53
3.4.13	Halaman <i>Add Userpanel</i> .....	54
3.4.14	Halaman <i>Edit Userpanel</i> .....	55
BAB V	PENUTUP .....	56
5.1	Kesimpulan .....	56
5.2	Saran .....	56
DAFTAR	PUSTAKA .....	lviii

## DAFTAR GAMBAR

<b>Gambar 1.1</b>	Statistik Jumlah <i>Webpage</i> Pornografi Didunia .....	1
<b>Gambar 2.1</b>	Layanan <i>Proxy Server</i> .....	6
<b>Gambar 2.2</b>	Skema Kinerja <i>Firewall</i> .....	7
<b>Gambar 2.3</b>	Ilustrasi Kerja <i>Domain Filtering</i> pada <i>Proxy</i> .....	10
<b>Gambar 3.1</b>	Sistem Kerja Standar <i>Proxy</i> .....	13
<b>Gambar 3.2</b>	Sistem Kerja <i>Proxy</i> Dengan Bantuan <i>Search engine</i> .....	14
<b>Gambar 3.3</b>	Desain Topologi Jaringan Komputer Dengan <i>Proxy</i> .....	14
<b>Gambar 3.4</b>	Desain Alur Data .....	15
<b>Gambar 3.5</b>	Sistem Kerja <i>Search Engine</i> dan Klien.....	16
<b>Gambar 3.6</b>	Alur <i>Squidsearch</i> Dengan Kolaborasi dengan <i>Search Engine</i> ..	16
<b>Gambar 3.7</b>	DFD Level 0 .....	19
<b>Gambar 3.8</b>	DFD Level 1 .....	20
<b>Gambar 3.9</b>	DFD Level 2 Proses 2 .....	21
<b>Gambar 3.10</b>	DFD Level 2 Proses 3 .....	21
<b>Gambar 3.11</b>	DFD Level 2 Proses 4.....	22
<b>Gambar 3.12</b>	DFD Level 2 Proses 5 .....	22
<b>Gambar 3.13</b>	DFD Level 2 Proses 6 .....	23
<b>Gambar 3.14</b>	Rancangan Tampilan Halaman <i>Login Squidsearch</i> .....	23
<b>Gambar 3.15</b>	Rancangan Halaman Utama <i>Squidsearch</i> .....	24
<b>Gambar 3.16</b>	Rancangan Halaman <i>Admin Manage Userpanel</i> .....	24
<b>Gambar 3.17</b>	Rancangan Halaman <i>Manage Keyword Search Squidsearch</i> ...	25
<b>Gambar 3.18</b>	Rancangan Halaman <i>Add Keyword Search Squidsearch</i> .....	25
<b>Gambar 3.19</b>	Rancangan Halaman <i>Manage Url Regex Squidsearch</i> .....	26
<b>Gambar 3.20</b>	Rancangan Halaman <i>Add Url Regex Squidsearch</i> .....	26
<b>Gambar 3.21</b>	Rancangan Halaman <i>Manage Blocked List Squidsearch</i> .....	27
<b>Gambar 3.22</b>	Rancangan Halaman <i>Add Blocked List Squidsearch</i> .....	27
<b>Gambar 3.23</b>	Rancangan Halaman <i>Proxy Setting Squidsearch</i> .....	28
<b>Gambar 4.1</b>	Halaman <i>Login Squidsearch</i> .....	42

<b>Gambar 4.2</b>	Halaman <i>Manage Url Regex</i> .....	42
<b>Gambar 4.3</b>	Halaman <i>Add Url Regex</i> .....	43
<b>Gambar 4.4</b>	Konfirmasi <i>Url Regex</i> .....	43
<b>Gambar 4.7</b>	Halaman <i>Edit Url Regex</i> .....	44
<b>Gambar 4.8</b>	Konfirmasi editing succesfull.....	45
<b>Gambar 4.9</b>	Halaman <i>Manage Keyword Search</i> .....	45
<b>Gambar 4.10</b>	Halaman <i>Add Keyword Search</i> .....	46
<b>Gambar 4.11</b>	Hasil Pencarian <i>Keyword Search</i> .....	46
<b>Gambar 4.13</b>	Gambar Laporan <i>Baned process request Error</i> .....	47
<b>Gambar 4.14</b>	Halaman <i>Edit Keyword Search</i> .....	48
<b>Gambar 4.15</b>	Konfirmasi <i>editing successful</i> .....	48
<b>Gambar 4.16</b>	Halaman <i>manage blocked list</i> .....	48
<b>Gambar 4.17</b>	Halaman <i>Add Blocked List</i> .....	49
<b>Gambar 4.18</b>	Konfirmasi <i>Add Url Successful</i> .....	49
<b>Gambar 4.19</b>	Halaman <i>Edit Blocked List</i> .....	50
<b>Gambar 4.20</b>	Konfirmasi <i>Successful Editing</i> .....	50
<b>Gambar 4.22</b>	Halaman <i>proxy Setting</i> .....	51
<b>Gambar 4.23</b>	Konfirmasi <i>squid start</i> .....	51
<b>Gambar 4.24</b>	Konfirmasi <i>squid stop</i> .....	52
<b>Gambar 4.25</b>	Konfirmasi <i>squid restart</i> .....	52
<b>Gambar 4.26</b>	Konfirmasi squid reconfigure.....	52
<b>Gambar 4.27</b>	Halaman <i>Automatic Update Squid</i> .....	53
<b>Gambar 4.28</b>	Halaman managemen <i>userpanel</i> .....	53
<b>Gambar 4.29</b>	Halaman <i>Add userpanel</i> .....	54
<b>Gambar 4.30</b>	Konfirmasi <i>Add userpanel successful</i> .....	54
<b>Gambar 4.31</b>	Halaman <i>edit userpanel</i> .....	55
<b>Gambar 4.32</b>	Konfirmasi <i>editing successful</i> .....	55

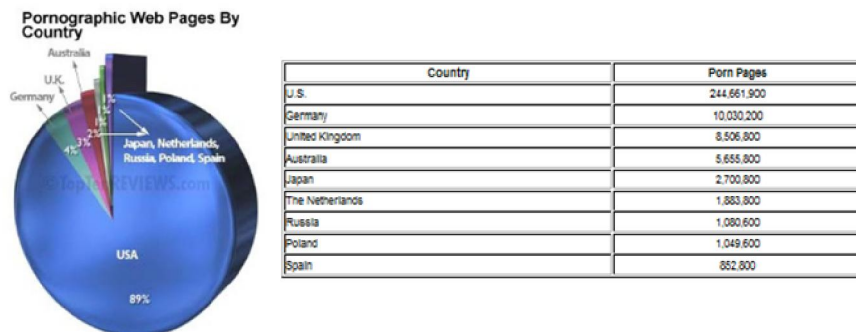
# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Dalam perkembangan teknologi informasi yang maju saat ini. Informasi sangat mudah didapatkan dengan cepat melalui teknologi *internet*, seperti *search engine*, portal berita, Wikipedia dsb. *Netizen* (pengguna *internet*) menggunakan *internet* sebagai cara untuk mendapatkan informasi dengan cepat. Namun, kadang tidak semua informasi dari *internet* tersebut kita dibutuhkan. Beberapa *website* bahkan berisi konten yang beresiko untuk keamanan komputer kita. Salah satunya adalah *website* yang mengandung pornografi yang sekarang ini sudah makin banyak.

Berdasarkan hasil survei yang dilakukan oleh badan sensus di Amerika Serikat didapatkan hasil statistik berupa jumlah *webpage website* porno yang ada di dunia. Dan sebagaimana ditunjukkan pada gambar 1.1.



**Gambar 1.1** Statistik Jumlah *Webpage* Pornografi Didunia ( sumber : *internet-filter-review.toptenreviews.com:2006*).

Dari hasil survei yang dilakukan pada tahun 2006 tersebut bisa dilihat bahwa beberapa negara memiliki *website* porno yang jumlahnya bahkan sudah jutaan, tentu saja hal ini dapat bertambah banyak lagi ke depannya nanti. Dari data di atas tentu saja kita membutuhkan teknologi yang dapat menyaring situs-situs tersebut. Mengingat *netizen* bukan hanya berasal dari rentang umur dewasa saja.



*Search engine* adalah alat pencari yang memudahkan *netizen* untuk mendapatkan akses dan informasi yang di inginkan dengan hasil yang cepat dan optimal. Mengingat *search engine* mempunyai teknologi yang dapat dalam menentukan *page rank* suatu halaman *website*. Salah teknologi yang mendukung adalah *semantic web* yang secara signifikan mempengaruhi hasil pencarian, bukan hanya mengacu pada kecocokan data, akan tetapi mengacu pada makna *keyword* yang dicari.

Salah satu fungsi *proxy server* adalah dapat membatasi akses klien ke jaringan *internet (web filtering)*. Seperti membatasi akses klien ke suatu *website* tertentu .Namun, pada penggunaannya proses pemasukan daftar situs yang dilarang ke basisdata, masih bersifat manual. Dan kurang efektif karena kita harus tahu dan selalu memonitor apakah ada *website* porno yang baru yang beredar di dunia maya dan memasukannya ke basisdata kita.

## 1.2 Rumusan Masalah

Bagaimana memanfaatkan hasil pencarian *search engine* agar dapat digunakan sebagai *update* daftar *website* yang dilarang didalam sistem *web filtering* secara otomatis.

## 1.3 Batasan Masalah

Adapun batasan masalah yang diangkat dalam penelitian ini sebagai berikut :

1. Penelitian difokuskan kepada cara untuk mendapatkan hasil pencarian *search engine* dan memasukannya ke basisdata *blocked list* sistem *proxy*.
2. Menggunakan *software squid* sebagai *Proxy Server* yang difungsikan sebagai *url filtering*.
3. Sistem menggunakan satu jenis *search engine*

4. Sistem hanya akan mengambil hasil pencarian sesuai dengan jumlah yang telah ditentukan.

#### **1.4 Tujuan Penelitian**

Merancang sebuah sistem yang dapat melakukan *update* basisdata *blocked list* secara otomatis berdasarkan hasil pencarian dari *search engine*.

#### **1.5 Manfaat Penelitian**

Memberikan kemudahan bagi *Administrator* jaringan untuk dapat meng-*update* basisdata sistem *proxy* secara otomatis.

#### **1.6 Metodologi Penelitian**

##### **1.6.1 Studi Pustaka**

Studi pustaka digunakan sebagai cara untuk mendapatkan informasi yang terkait dengan penelitian yaitu buku-buku dan *internet*, untuk penelitian yang dilakukan mencakup implementasi penggunaan *search engine* serta beberapa hal yang berkaitan dengan basisdata MySQL dan pengintegrasian dengan *web proxy filtering* yang dalam hal ini adalah *squid proxy*. Dari studi pustaka diharapkan landasan teori atau metode untuk pengembangan *Automatic update web filtering* berbasis *squid* dengan bantuan *search engine*

##### **1.6.2 Implementasi dan Konfigurasi Sistem**

Pengembangan sistem ini disusun dari hasil beberapa hasil studi pustaka meliputi :

- a. Desain Arsitektur Jaringan

Tahap ini merupakan tahap perancangan topologi jaringan komputer lokal yang akan digunakan untuk klien dan *squid Proxy Server* sebagai *web Filtering*. Pengujian untuk *blocked website* akan menggunakan *website* yang ada di *internet*.

b. Desain Alur Proses Aplikasi

Tahap ini merupakan perancangan bagaimana data mengalir pada setiap proses dari awal sampai akhir yang terjadi pada sistem *squidsearch*.

c. Pengadaan Perangkat Keras

Tahap ini merupakan tahap pengadaan perangkat keras yang di butuhkan dalam pembuatan sistem. Baik berupa *Server* yang akan di gunakan sebagai *Gateway*, maupun komputer klien.

d. Instalasi dan Konfigurasi Sistem

Tahap ini merupakan tahap instalasi *software* yang dibutuhkan dalam pembuatan sistem dan pengkonfigurasi sistem sesuai kebutuhan. *Software* yang di butuhkan pada *Gateway* adalah. *Apache Server*. *Mysql Server* dan *squid proxy Server*

e. Pengujian

Tahap ini merupakan tahap terakhir setelah sistem *web filtering* yang ada telah di buat. Skenario pengujian dimulai saat *Administrator* akan membuka sistem *web filtering*. Kemudian *Administrator* memasukan *keyword* tertentu. Hasil pencarian dari *search engine* akan dimasukan kedalam basisdata, dan kita bisa melihat dampaknya pada daftar *website* yang di blok akan bertambah sesuai dengan hasil pencarian *search engine*.

## 1.7 Sistematika Penulisan

### BAB I PENDAHULUAN

Membahas tentang latar belakang masalah, batasan masalah, rumusan masalah, tujuan penelitian serta manfaat dari penelitian dan metodologi penelitian

### BAB II LANDASAN TEORI

Membahas dasar teori yang digunakan dalam perancangan dan pembangunan sebuah *proxy* dengan fungsi tambahan *update* daftar *website* secara otomatis

### BAB III METODOLOGI

Memuat uraian tentang analisis masalah, gambaran umum sistem, analisis kebutuhan sistem yang mencakup kebutuhan perangkat keras, perangkat jaringan, dan perangkat lunak yang digunakan untuk membantu penyelesaian tugas akhir.

### BAB IV HASIL PEMBAHASAN

Memuat tahap-tahap instalasi, konfigurasi serta pembuatan sistem dan pengujian terhadap sistem. Yaitu dengan memasukan *keyword* pencarian ke sistem dan melihat keluaran dari *search engine*. Kemudian melihat dampaknya pada basisdata sistem *proxy*.

### BAB V PENUTUP

Memuat kesimpulan dan juga membahas saran yang dapat digunakan oleh pihak yang berkepentingan maupun untuk peneliti terhadap kekurangan serta keterbatasan dalam penelitian ini.

## BAB II

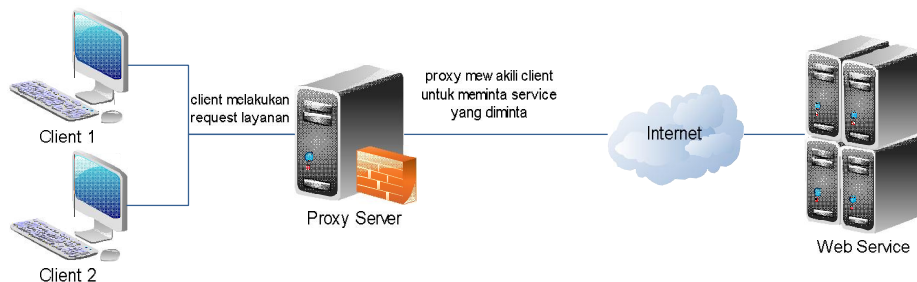
### LANDASAN TEORI

#### 2.1 Konsep Kerja *Filtering Proxy*

*Proxy* dapat dipahami sebagai pihak ketiga yang berdiri di tengah-tengah antara kedua pihak yang saling berhubungan dan berfungsi sebagai perantara. *Proxy Server* mempunyai 3 fungsi utama yaitu *Connection Sharing*, *Filtering* dan *Caching*.

Dalam suatu jaringan lokal yang terhubung ke jaringan lain atau *internet*, pengguna tidak langsung berhubungan dengan jaringan luar atau *internet*, tetapi harus melewati suatu *Gateway*, yang bertindak sebagai batas antara jaringan lokal dan jaringan luar. *Gateway* ini sangat penting, karena jaringan lokal harus dapat dilindungi dengan baik dari bahaya yang mungkin berasal dari *internet*, dan hal tersebut akan sulit dilakukan bila tidak ada garis batas yang jelas jaringan lokal dan *Internet*. Dalam hal ini, *Proxy Server* dapat bertindak sebagai *Gateway*, karena bertindak sebagai perantara antara jaringan lokal dan jaringan luar atau *Internet*.

*Proxy* juga bekerja sebagai perantara *request* klien terhadap layanan yang tersedia di dalam *Internet*, sebagai contoh yaitu : *http service*. Pada saat klien melakukan *request* terhadap suatu halaman *website*, maka yang terjadi adalah *proxy* akan mewakili *request* tersebut ke *internet*, dan saat halaman sudah di terima *proxy* maka halaman tersebut akan diteruskan ke klien. Seperti ditunjukkan pada Gambar 2.1.



**Gambar 2.1** Layanan *Proxy Server*

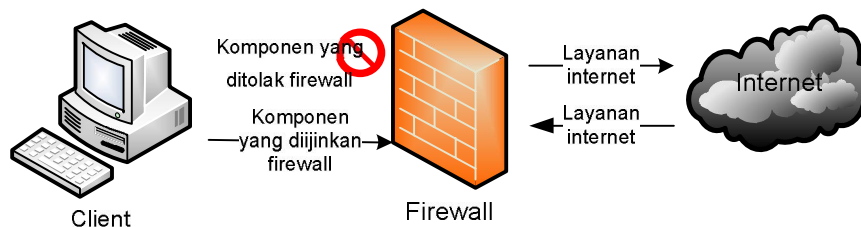
### 2.1.1 Gateway

Untuk dapat mengakses *internet*, sebuah komputer harus memiliki sebuah *ip public*. Hal ini menjadi mustahil karena IPv4 yang tersedia di dunia sudah habis, sehingga untuk dapat mengakses *Internet* secara bersama-sama dengan menggunakan satu *ip public*, dibutuhkan sebuah komputer yang memiliki *ip public*, yang digunakan sebagai *gateway* untuk komputer lain.

Dalam hal ini *proxy server* juga berfungsi sebagai *Gateway*. *Server* ini mempunyai dua *interface*, satu untuk antarmuka dengan *internet* dan satu untuk antarmuka dengan jaringan lokal.

### 2.1.2 Firewall

*Firewall* adalah sebuah alat atau sistem yang berfungsi mengelola keamanan suatu jaringan dari serangan atau ancaman dari luar. Beberapa ancaman dari luar misalnya *virus*, *malware*, halaman *website* tertentu, bahkan peretas yang akan masuk ke jaringan dapat terhalang oleh *firewall*. *firewall* terletak diantara klien dan *server* (*internet*) seperti gambar berikut:



**Gambar 2.2** Skema Kinerja *Firewall*

Gambar 2.2 menjelaskan fungsi utama *firewall* adalah mengontrol dan mengatur lalu lintas jaringan. Aturan yang dimiliki *firewall* yaitu: diijinkan atau ditolak. Komponen yang diijinkan akan diteruskan oleh *firewall*, sedangkan komponen yang ditolak akan langsung dibuang. Aturan *firewall* meliputi seluruh komponen jaringan seperti *ip address*, *port*, *domain*, *protocol* dan sebagainya. Aturan yang diberikan kepada *Firewall* disimpan dalam sebuah daftar

aturan yang disebut *Access Control Lists (ACL)*. Aturan *firewall* terhadap komponen jaringan didefinisikan secara manual oleh *Administrator*.

*Firewall* bekerja pada beberapa *layer* jaringan. *Firewall* yang bekerja pada *layer* jaringan kedua dapat mengelola paket berdasar tipe jaringan, misalnya ethernet. *Packet filtering firewall* bekerja pada *layer* ketiga (*network*). *Firewall* pada *layer* ketiga dapat mengelola paket berdasarkan alamat IP. *Firewall* pada *layer* ke empat dapat mengelola paket berdasarkan *port*, misalnya *port* 80 untuk HTTP. *Firewall* pada *layer* ketujuh atau *application-level gateway firewall* seperti yang terdapat pada *proxy* mampu mengelola *domain*, waktu, autentikasi user, dan sebagainya.

### **1. Packet Filtering**

Fungsi paling sederhana dari *firewall* adalah melakukan *packet filtering firewall*. Paket data adalah komponen berisi data atau informasi digital yang ditransmisikan melalui jaringan komputer. Tiap paket data memiliki *header* yang berisi informasi penting seperti *ip address*, nama protokol, nomor *port*, alamat *domain*, ukuran paket dan sebagainya. *Packet filtering firewall* bekerja pada *layer* ketiga OSI.

*Packet filtering firewall* memeriksa atau menyaring data yang masuk dalam jaringan, membandingkannya dengan aturan dalam ACL, kemudian memutuskan untuk menolak atau mengizinkan paket tersebut. Jika paket termasuk daftar yang ditolak maka paket akan dihentikan, jika termasuk daftar yang diizinkan paket akan diteruskan ke jaringan. Semakin banyak aturan pada *firewall* termasuk pengecualian yang diberlakukan, maka ukuran ACL juga akan membesar. *Packet filtering firewall* banyak diimplementasikan pada perangkat jaringan router yang dapat

memutuskan akan meneruskan paket ke tujuan atau menolaknya. Aplikasi *packet filtering firewall* misalnya *iptables* pada linux.

*Packet filtering firewall* tidak memerlukan konfigurasi tertentu mengenai pengguna seperti autentikasi *user*. Aturan pada *packet filtering firewall* hanya mengizinkan atau menolak paket yang lewat berdasarkan *header* paket datanya, karena itu *packet filtering firewall* mampu bekerja cepat dan tidak memerlukan spesifikasi *hardware* yang tinggi. Kelemahan *packet filtering firewall* ada pada daftar aturan yang akan semakin banyak dan kompleks seiring bertambahnya aturan yang diberlakukan.

## 2. Domain Filtering

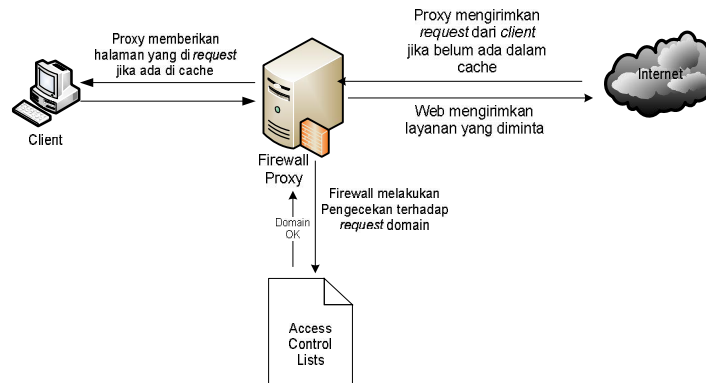
*Domain* merupakan nama yang digunakan untuk representasi alamat IP pada jaringan dalam bentuk bahasa manusia. Selain beberapa servis seperti FTP, *domain* banyak digunakan untuk penamaan sebuah *website*. *Domain* itu sendiri dikelola oleh DNS (*Domain Name system*) *Server* yang berfungsi menerjemahkan nama *domain* ke alamat IP dan sebaliknya. DNS *Server* bekerja pada *layer* ketujuh (aplikasi) model jaringan OSI, sama seperti *proxy*. *Proxy* memiliki fungsi sebagai *Firewall* yang bekerja dengan prinsip *domain filtering*.

*Domain Filtering* merupakan penyaringan alamat *web* atau *domain* secara keseluruhan maupun sebagian. *Domain Filtering* bekerja dengan memeriksa URL yang di *request* oleh klien, membandingkannya dengan aturan yang berlaku kemudian memutuskan untuk mengizinkan atau menolak akses. *Domain filtering* berbeda dengan *firewall* biasa, karena *firewall* biasa hanya dapat membaca *filtering firewall* paket data, bukan *domain*. *Proxy* bekerja pada *layer* ketujuh jaringan komputer, lebih tinggi dari *firewall* biasa maupun router dengan fitur paket *filtering*. Hal ini membuat *firewall* pada *proxy* memiliki lebih banyak fungsi dan



kontrol terhadap akses jaringan sehingga dapat melakukan *domain Filtering*. *Web Proxy Server* dapat melihat URL dari halaman *web* yang di *request* klien dengan memeriksa pesan HTTP, GET dan POST. Dengan kemampuan tersebut *firewall* pada *proxy* dapat menolak atau mengizinkan akses ke *domain* tertentu. Kebijakan dan aturan pada *domain filtering* sama dengan *firewall* pada umumnya, yaitu mengizinkan atau menolak akses.

Seperti *firewall* lain, *proxy firewall* juga memiliki *Access Control Lists (ACL)* yang berisi aturan-aturan *domain filtering*. Berikut ini gambaran kinerja *firewall* pada *proxy*:



**Gambar 2.3** Ilustrasi Kerja *Domain Filtering* pada *Proxy*

Berbeda dengan *packet filtering firewall* yang tidak memerlukan spesifikasi *hardware* tinggi, *domain Filtering* memiliki kompleksitas kinerja tinggi sehingga membutuhkan sumber daya yang besar. *Domain Filtering* mampu mengelola autentikasi *user* dan kontrol akses, sehingga akses jaringan lebih aman (John, 2002:3). *Domain Filtering* bekerja berdampingan dengan *proxy* sehingga diperlukan aplikasi yang mendukung serta spesifikasi sumber daya yang tinggi agar *proxy* dapat berjalan dengan baik. *Domain Filtering* bekerja dengan sebuah file yang berisi daftar alamat *domain*. (Nugroho,2011:13)

## 2.2 *Search engine*

*Search engine* adalah istilah yang digunakan bagi *website* yang berfungsi sebagai mesin pencari, *Search engine* ini akan menampilkan informasi yang diminta oleh *user* berdasarkan permintaan yang di masukkan pada suatu *form* pencari konten, konten yang ditampilkan adalah konten yang memang sudah terindex dan tersimpan di *Server search engine* tersebut.

*Search engine* berisi sistem basisdata yang dirancang untuk mengindex alamat-alamat *website* di *internet* . Untuk melaksanakan tugasnya ini, *search engine* memiliki program khusus yang biasanya disebut *spider crawler*. Pada saat Anda mendaftarkan sebuah alamat *website* (URL), *spider* dari *search engine* tersebut akan menerima dan menganalisis URL tersebut lalu meng-crawl-nya.

Dengan proses dan prosedur yang serba otomatis, *spider* ini akan memutuskan apakah *web* yang anda daftarkan layak diterima atau tidak pada *search engine* tersebut. Jika layak, *spider* akan langsung menambahkan alamat URL tersebut ke sistem basisdata mereka. Rangka-pun segera ditetapkan dengan algoritma dan caranya masing-masing. Jika tidak layak, terpaksa Anda harus bersabar dan mengulangi pendaftaran diwaktu yang lain. Dan program inilah yang sebenarnya menentukan posisi suatu *web* site di *search engine* tersebut. Contoh beberapa *search engine* terkenal adalah : Altavista, Google, Excite, Northern Light, Hotbot, yahoo dan masih banyak yang lainnya.

*Semantic search* adalah sebuah teknologi yang berfokus pada makna kata dalam melakukan pencarian. Teknologi ini menggunakan *artificial intelligence* untuk memahami kemampuan pencari dan hubungan serta makna kata kunci pencarian.

Karenanya hasil pencarian nantinya akan lebih baik lagi dibandingkan dengan yang sudah ada sekarang ini. *Search engine* akan mengerti bahwa

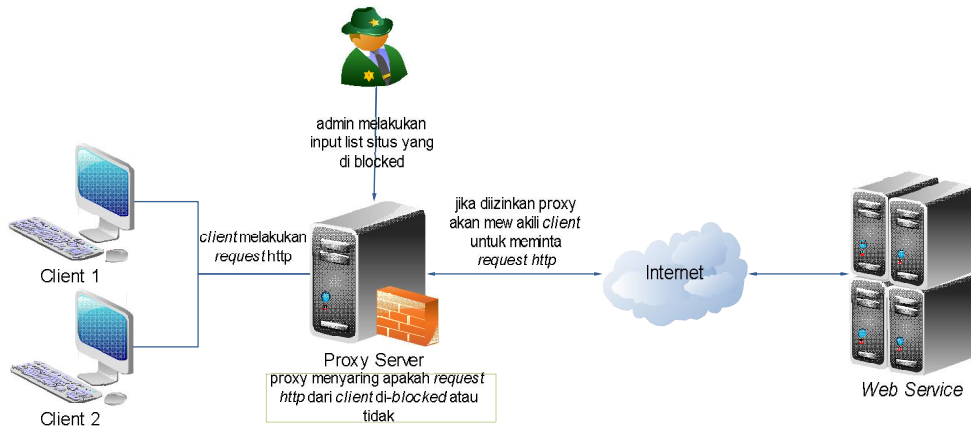
“*their*” dan “*they’re*” berbeda dan penggabungan kata “*new*” dan “*York*” akan mengubah makna.

Saat ini proses pencarian didasarkan pada *keyword*. Mesin pencari akan memberikan hasil pencarian dengan mencocokkan kata dengan hasil seperti bila kita mencari melalui kamus. Singkatnya, saat ini *Search engine* memberikan hasil sesuai terkaan terbaiknya.

*Semantic search* tidak hanya soal menempatkan kata kunci di tempat yang tepat, tetapi juga konten yang relevan dengan pencarian. Teknologi ini lebih mementingkan makna di balik kata kunci pencarian, bukan semata-mata mencari kecocokan kata. Karenanya mesin pencari ini akan mencari informasi dan data yang memang relevan. Bila konten Anda tidak relevan dengan apa yang dicari, maka *website* Anda tidak akan muncul dalam hasil pencarian meski di dalamnya bertebaran banyak kata kunci.

## BAB III METODOLOGI

### 3.1 Analisis Masalah

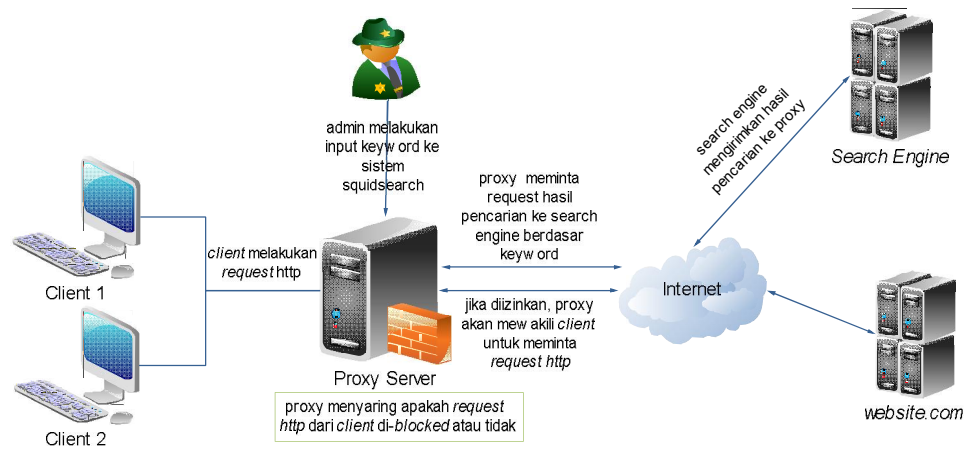


**Gambar 3.1** Sistem Kerja Standar *Proxy*

Untuk dapat melakukan *bloking* akses terhadap URL *website* tertentu, salah satu cara yang dapat dilakukan adalah dengan mengkonfigurasi *Proxy Server* sebagai penghubung klien dengan internet seperti yang ditunjukkan pada gambar 3.2. Pada dasarnya, *proxy* melakukan pelarangan akses terhadap url *website* menggunakan basisdata yang berisi data berupa URL *website* yang telah dideklarasikan sebelumnya. basisdata yang dimaksudkan disini adalah file konfigurasi *proxy* itu sendiri atau file eksternal yang sudah didefinisikan pada file konfigurasi *proxy*.

Penyimpanan daftar *website* yang berbentuk file inilah yang sangat menyulitkan *Administrator* untuk proses peng-upate-an. *Administrator* akan selalu melakukan input daftar *website* yang di blok kedalam sebuah file. Hal ini tentu saja kurang efektif untuk *Administrator*.

Berdasarkan *Analisis* yang telah di jabarkan diatas, maka diperlukan sebuah sistem yang lebih mudah dan dapat melakukan membantu *update blocked list* secara otomatis. Salah satunya adalah dengan memadukan teknologi *Filtering Proxy* dengan teknologi *Search Engine*.



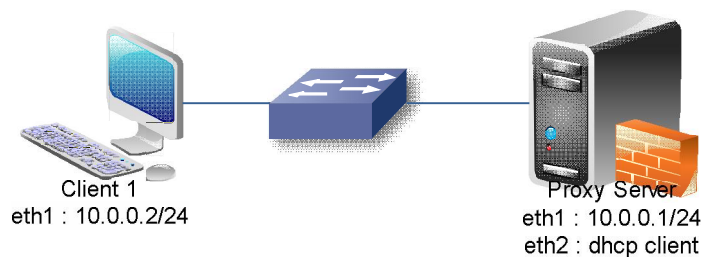
**Gambar 3.2** Sistem Kerja *Proxy* Dengan Bantuan *Search engine*

Gambar 3.2 di atas merupakan desain *proxy* apabila ditambahkan fungsi otomatis *update* dalam teknologi *search engine*. Dengan cara ini *Administrator* dipermudah untuk dapat memasukan dan meng-*update* basisdata *blocked list* yang ada pada *proxy*.

Sistem dibuat untuk dapat berinteraksi dengan *search engine* agar dapat mendapatkan hasil pencarian berdasarkan pada keyword yang administrator masukan. Dan setelahnya *Proxy Server* akan berkolaborasi dengan *search engine* agar hasil pencarian dapat di masukan kedalam basisdata *blocked list*.

### 3.2 Gambaran Umum Sistem

Implementasi mekanisme jaringan yang digunakan adalah jaringan lokal.



**Gambar 3.3** Desain Topologi Jaringan Komputer Dengan *Proxy*

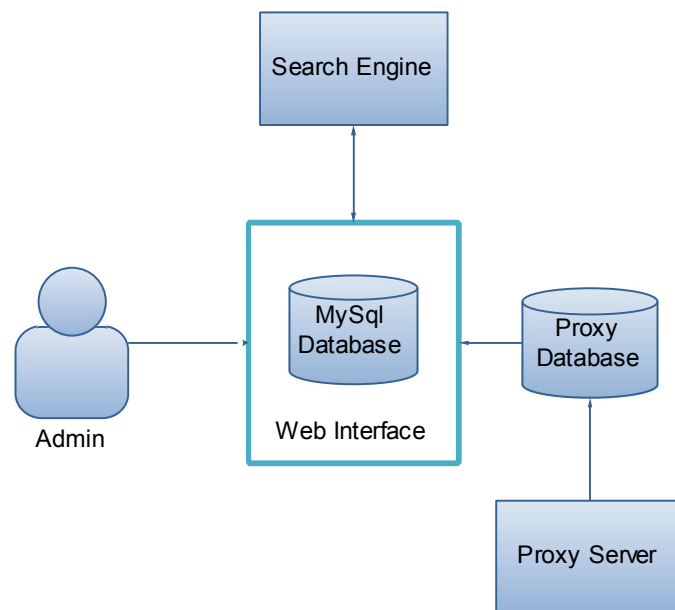
Dalam jaringan lokal terdapat satu *server* yang dalam hal ini bertindak sebagai *proxy Server*, dan satunya bertindak sebagai klien. Topologi yang digunakan adalah seperti pada gambar 3.3. Dalam topologi tersebut

terdapat dua jaringan yang akan di gunakan, yaitu jaringan lokal yang digunakan untuk terhubung dengan klien. Dan jaringan *internet* dimana *proxy* akan terhubung dengan *search engine*. Dan berikut keterangan lebih jelasnya :

1. *Eth1* : merupakan *interface* yang digunakan untuk *proxy* terkoneksi dengan *internet* dan diseting sebagai dhcp client.
2. *Eth2* : merupakan interface yang terhubung dengan jaringan lokal dengan *ip static*.

Komputer klien di sini bertindak sebagai penguji sistem. Komputer ini hanya mempunyai satu *interface* untuk terkoneksi ke *proxy*. Dalam sistem *web filtering* kita tidak menggunakan konfigurasi *transparent proxy*. Sehingga klien harus diarahkan untuk dapat menggunakan *service proxy*

### 3.2.1 Desain dan Alur Data Proses Squidsearch

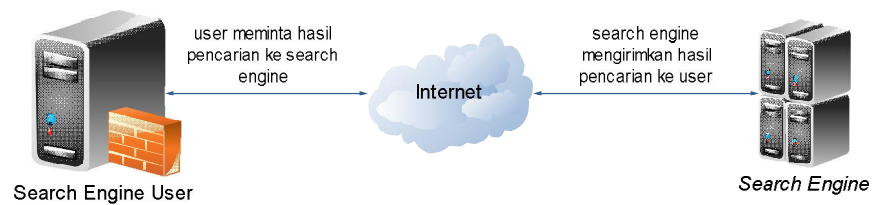


**Gambar 3.4** Desain Alur Data

Gambar 3.4 merupakan penjelasan proses alur data yang akan terjadi pada saat sistem *squidsearch* terbentuk. Didalam sistem *proxy* akan di sediakan suatu *web interface* dimana *Administrator* dapat mengkonfigurasi *squidsearch* dengan mudah.

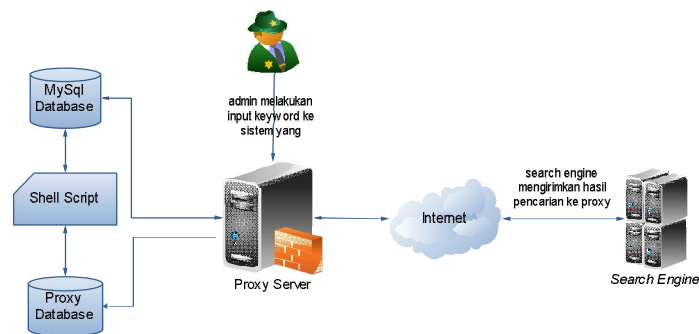
### 3.2.2 Cara Kerja Sistem *Search Engine*

Pada setiap *search engine* akan disediakan suatu *form* yang berguna sebagai tempat memasukkan *keyword* yang ingin dicari. Saat *keyword* sudah dimasukan maka *search engine* akan mencari dibasisdata *index* mereka dan menampilkan hasil dalam bentuk daftar alamat *website* yang mengandung *keyword* tersebut. Seperti ditunjukkan pada gambar 3.5.



**Gambar 3.5** Sistem Kerja *Search Engine* dan Klien

Dengan *search engine* kita dapat mencari informasi yang kita inginkan. Bahkan dengan parameter-parameter tertentu sesuai dengan kebutuhan kita. Dimulai pada saat klien membuka *search engine* dan memasukan *keyword* yang dicari. *Request* hasil pencarian akan di terima *search engine*. *Request* yang diterima oleh *search engine* selanjutnya akan diproses sesuai dengan parameter yang kita masukan. *Search engine* juga mempunyai suatu basisdata *index* tentang informasi yang ada dan alamatnya. menjadi hasil pencarian berupa url-url yang berhubungan dengan *keyword* yang di inputkan.



**Gambar 3.6** Alur *Squidsearch* Dengan Kolaborasi dengan *Search Engine*

*Search engine* pada dasarnya memberikan hasil pencarian berdasar *page rank* sehingga cukup baik kalau kita masukan sebagai acuan *proxy* dalam melakukan pelarangan.

*Temporary* basisdata yang dalam hal ini adalah basiswata Mysql berfungsi sebagai penyimpanan data Domain yang akan diblok. Sedangkan *shell script* adalah program yang dibuat menggunakan bahasa *shell*. Program ini dimaksudkan untuk memasukan data yang ada pada basisdata Mysql ke dalam basisdata *proxy* yang berbentuk file.

### 3.3 Analisis dan persiapan kebutuhan sistem

#### 3.3.1 Kebutuhan perangkat keras

Berdasarkan gambaran umum sistem, terlihat bahwa perangkat-perangkat keras yang dibutuhkan untuk menunjang implementasi dari semua fitur-fitur yang ada pada sistem adalah sebagai berikut:

##### 1. Komputer sebagai *Squidsearch*

Dalam komputer ini,terdapat squid *Proxy Server* dan *web Server* yang akan di gunakan sebagai tempat aplikasi sistem. Spesifikasi yang digunakan adalah :

- a. Prosesor 1.6 GHz
- b. Memori 512 MB
- c. *Harddisk* dengan kapasitas 80 GB
- d. *Network Interface Card* 2 buah

##### 2. Komputer sebagai klien

Komputer yang digunakan untuk pengujian.

- a. Prosesor 1.6 GHz
- b. Memori 2 GB
- c. *Harddisk* dengan kapasitas 160 GB
- d. *Network Interface Card* 1 buah



### 3.3.2 Kebutuhan perangkat lunak

#### 1. Apache2

Apache adalah sebuah aplikasi yang digunakan untuk membangun sebuah *web server protocol* yang digunakan untuk melayani fasilitas *web* menggunakan HTTP. *Web Server* berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan *web browser* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen HTML Apache merupakan perangkat lunak *open source* yang dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation*.

#### 2. Mysql

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL. MySQL berfungsi sebagai media penyimpanan data yang bersifat dinamis dan disimpan dalam tabel-tabel yang tersedia di dalamnya. MySQL merupakan implementasi dari *Relational basisdata Management Sistem (RDBMS)* yang didistribusikan secara gratis dibawah lisensi *General Public License (GPL)*. *Structured Query Language (SQL)* adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah.

#### 3. Squid

*Squid* adalah sebuah aplikasi yang digunakan sebagai *proxy web filtering* dan *web cache*. Squid memiliki banyak jenis penggunaan, mulai dari mempercepat *web server* dengan melakukan *caching* permintaan yang berulang-ulang, *caching DNS*, *caching website web*, dan *caching* pencarian komputer di dalam jaringan untuk sekelompok komputer yang menggunakan sumber daya jaringan yang sama, hingga membantu keamanan

dengan cara melakukan penyaringan lalu lintas data. *Squid* umumnya dirancang untuk berjalan diatas sistem operasi yang berbasis UNIX.

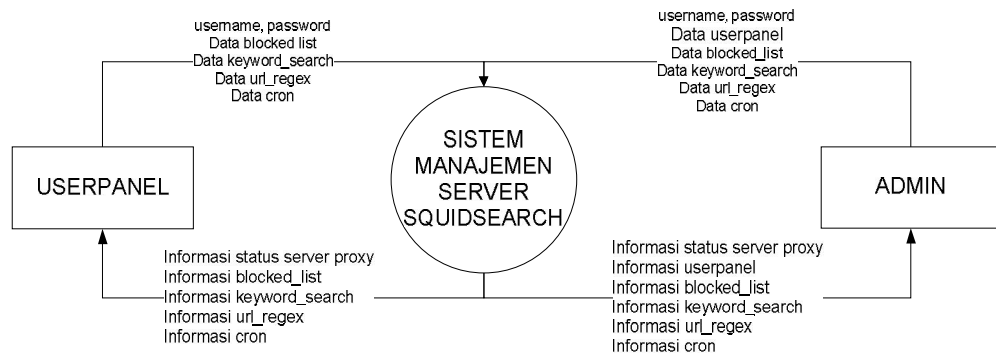
#### 4. Cron

Cron adalah sebuah aplikasi yang digunakan untuk penjadwalan proses yang ada dalam linux. Secara otomatis akan melakukan proses yang sudah dimasukkan kedalam daftar tugas aplikasi cron.

### 3.4 Perancangan Aplikasi *Squidsearch*

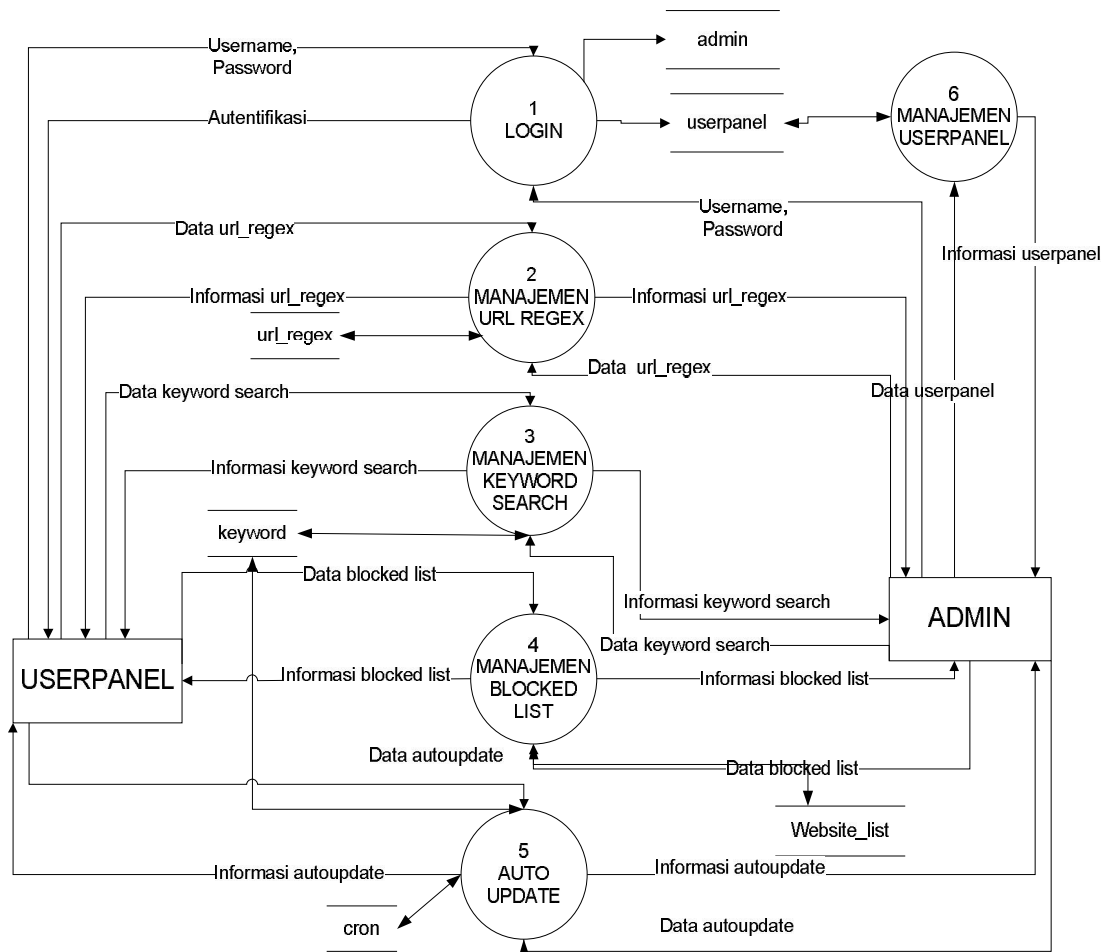
Perancangan aplikasi direpresentasikan melalui *Data Flow Diagram* (DFD), berikut perancangan aplikasi *Squidsearch* :

#### 3.4.1 DFD Level 0



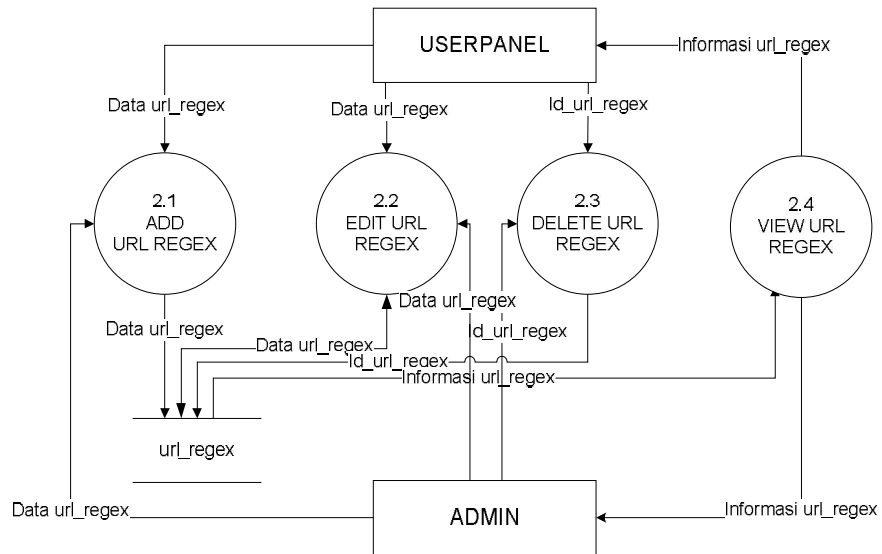
**Gambar 3.7** DFD Level 0

## 3.4.2 DFD Level 1



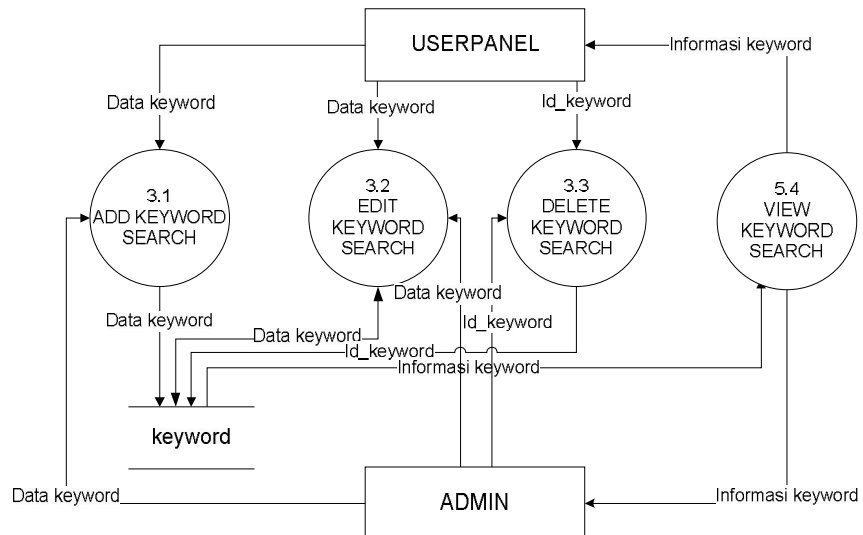
Gambar 3.8 DFD Level 1

### 3.4.3 DFD Level 2 Proses 2



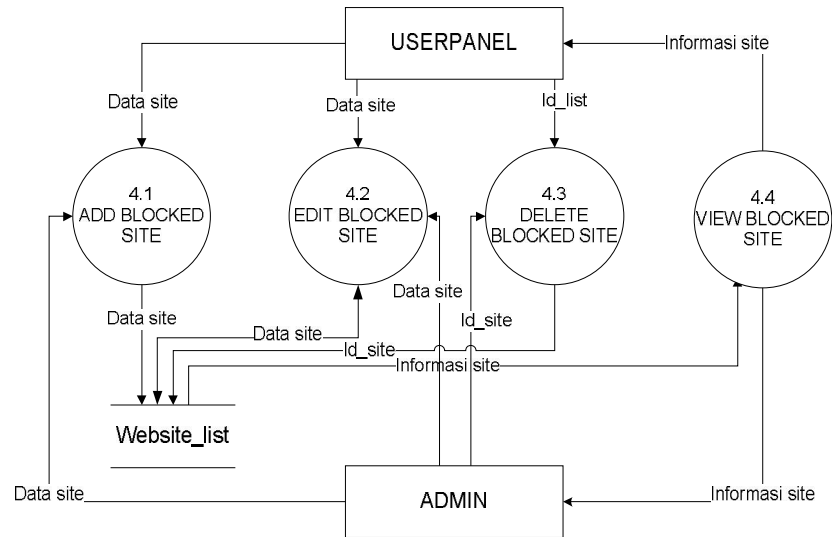
**Gambar 3.9** DFD Level 2 Proses 2

### 3.4.4 DFD Level 2 Proses 3



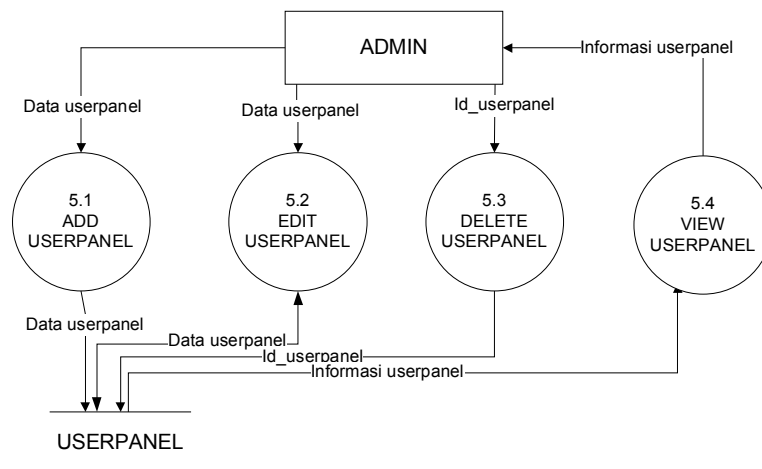
**Gambar 3.10** DFD Level 2 Proses 3

### 3.4.5 DFD Level 2 Proses 4



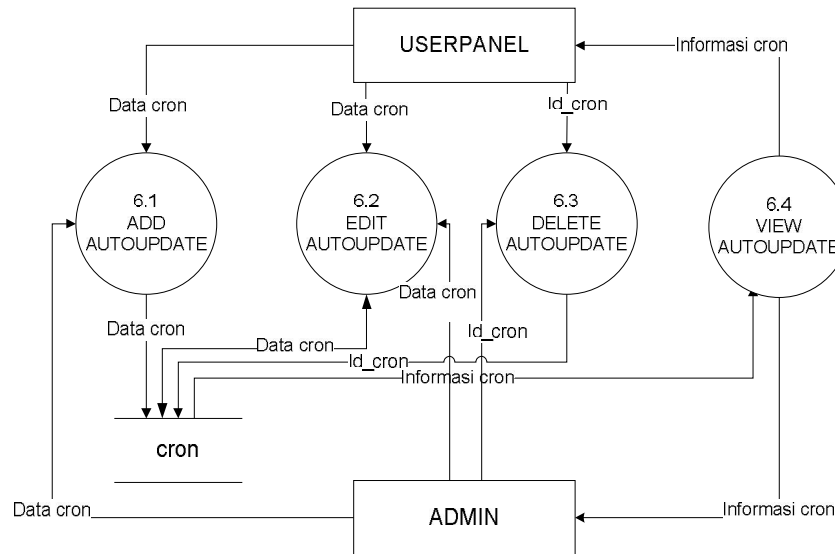
**Gambar 3.11** DFD Level 2 Proses 4

### 3.4.6 DFD Level 2 Proses 5



**Gambar 3.12** DFD Level 2 Proses 5

### 3.4.7 DFD Level 2 Proses 6



**Gambar 3.13** DFD Level 2 Proses 6

## 3.5 Perancangan Tampilan *Squidsearch*

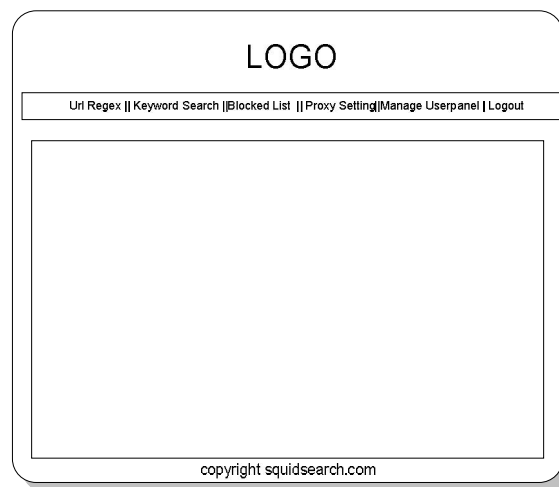
### 3.5.1 Tampilan Halaman *Login Squidsearch*

Gambar 3.14 merupakan rancangan halaman login squidsearch, dimana userpanel/admin harus memasukan username dan password sebelum masuk kedalam sistem squidsearch.

**Gambar 3.14** Rancangan Tampilan Halaman *Login Squidsearch*

### 3.5.2 Tampilan Halaman Utama *Squidsearch*

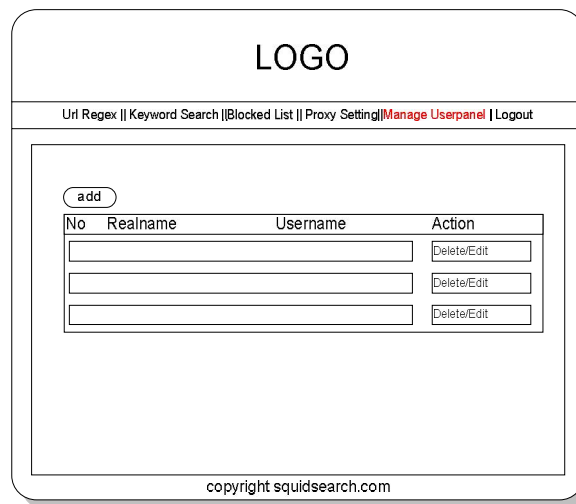
Gambar 3.15 merupakan rancangan halaman utama *squidsearch*. Halaman ini merupakan halaman yang muncul saat login telah berhasil.



**Gambar 3.15** Rancangan Halaman Utama *Squidsearch*

### 3.5.3 Tampilan Halaman *Admin Manage Userpanel*

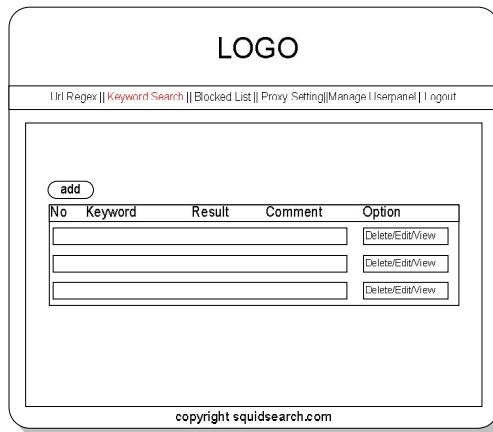
Gambar 3.16 merupakan rancangan halaman *Admin Manage Userpanel*, dihalaman ini *admin* dapat melakukan manajemen *userpanel*.



**Gambar 3.16** Rancangan Halaman *Admin Manage Userpanel*

### 3.5.4 Tampilan Halaman *Manage Keyword Search Squidsearch*

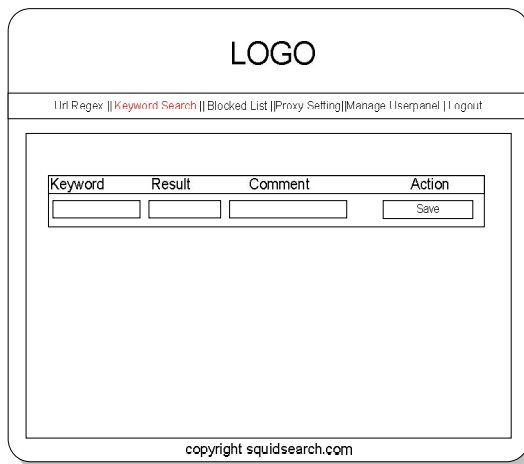
Gambar 3.17 merupakan rancangan halaman *Manage Keyword Search Squidsearch*. Di halaman ini userpanel dapat melakukan manajemen *keyword search*.



**Gambar 3.17** Rancangan Halaman *Manage Keyword Search Squidsearch*

### 3.5.5 Tampilan Halaman *Add Keyword Search Squidsearch*

Gambar 3.18 merupakan rancangan halaman *Add Keyword Search Squidsearch*. Di halaman ini *userpanel* dapat memasukkan *keyword search* yang digunakan sistem untuk meminta hasil pencarian ke *search engine*.

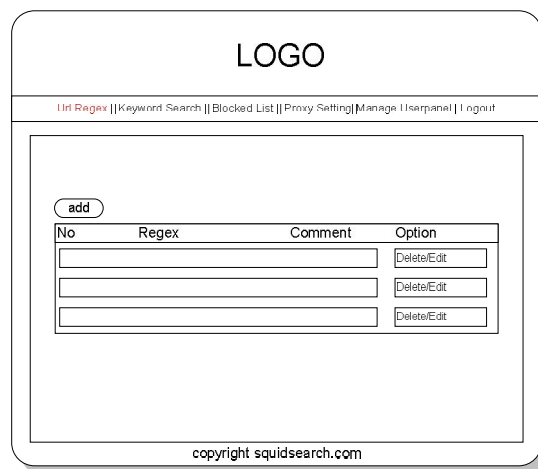


**Gambar 3.18** Rancangan Halaman *Add Keyword Search Squidsearch*



### 3.5.6 Tampilan Halaman *Manage Url Regex Squidsearch*

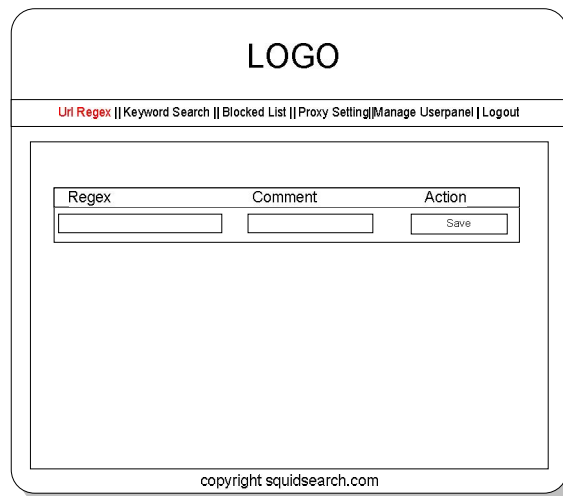
Gambar 3.19 merupakan rancangan halaman *Manage Url Regex Squidsearch*. Di halaman ini *userpanel* dapat melakukan manajemen *Url Regex*.



**Gambar 3.19** Rancangan Halaman *Manage Url Regex Squidsearch*

### 3.5.7 Tampilan Halaman *Add Url Regex Squidsearch*

Gambar 3.20 merupakan rancangan halaman *Add Url Regex Squidsearch*. Di halaman ini *userpanel* dapat memasukan *regex* yang diinginkan.



**Gambar 3.20** Rancangan Halaman *Add Url Regex Squidsearch*

### 3.5.8 Tampilan Halaman *Manage Blocked List Squidsearch*

Gambar 3.21 merupakan rancangan halaman *Manage Blocked List Squidsearch*. Di halaman ini *userpanel* dapat melakukan *managemen Blocked List*.



The image shows a wireframe for the 'Manage Blocked List' page. At the top, there is a 'LOGO' placeholder. Below it is a navigation bar with links: 'Url Regex', 'Keyword Search', 'Blocked List' (highlighted in red), 'Proxy Setting', 'Manage Userpanel', and 'Logout'. The main content area contains a table with the following structure:

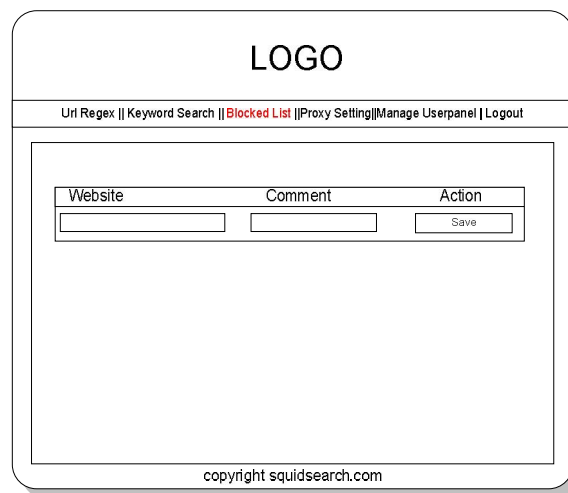
No	Website	Comment	Action
			Delete/Edit

At the bottom of the page, there is a copyright notice: 'copyright squidsearch.com'.

**Gambar 3.21** Rancangan Halaman *Manage Blocked List Squidsearch*

### 3.5.9 Tampilan Halaman *Add Blocked List Squidsearch*

Gambar 3.22 merupakan rancangan halaman *Add Blocked List Squidsearch*. Di halaman ini *userpanel* dapat memasukan *url website* yang diinginkan untuk dilarang aksesnya.



The image shows a wireframe for the 'Add Blocked List' page. At the top, there is a 'LOGO' placeholder. Below it is a navigation bar with links: 'Url Regex', 'Keyword Search', 'Blocked List' (highlighted in red), 'Proxy Setting', 'Manage Userpanel', and 'Logout'. The main content area contains a form with the following structure:

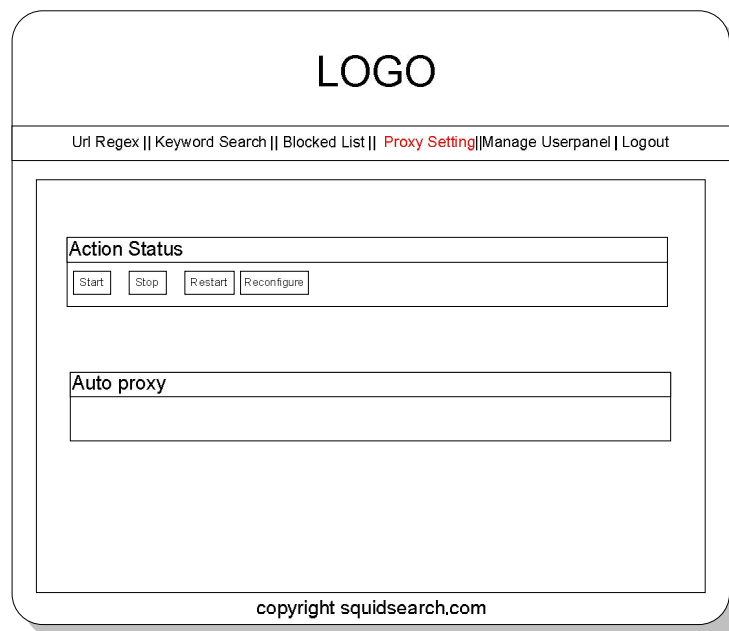
Website	Comment	Action
<input type="text"/>	<input type="text"/>	Save

At the bottom of the page, there is a copyright notice: 'copyright squidsearch.com'.

**Gambar 3.22** Rancangan Halaman *Add Blocked List Squidsearch*

### 3.5.10 Tampilan Halaman *Proxy Setting Squidsearch*

Gambar 3.23 merupakan rancangan halaman *Proxy Setting Squidsearch*. Di halaman ini *userpanel* dapat mengatur *automatic update*, dan mengatur *daemon squid*.

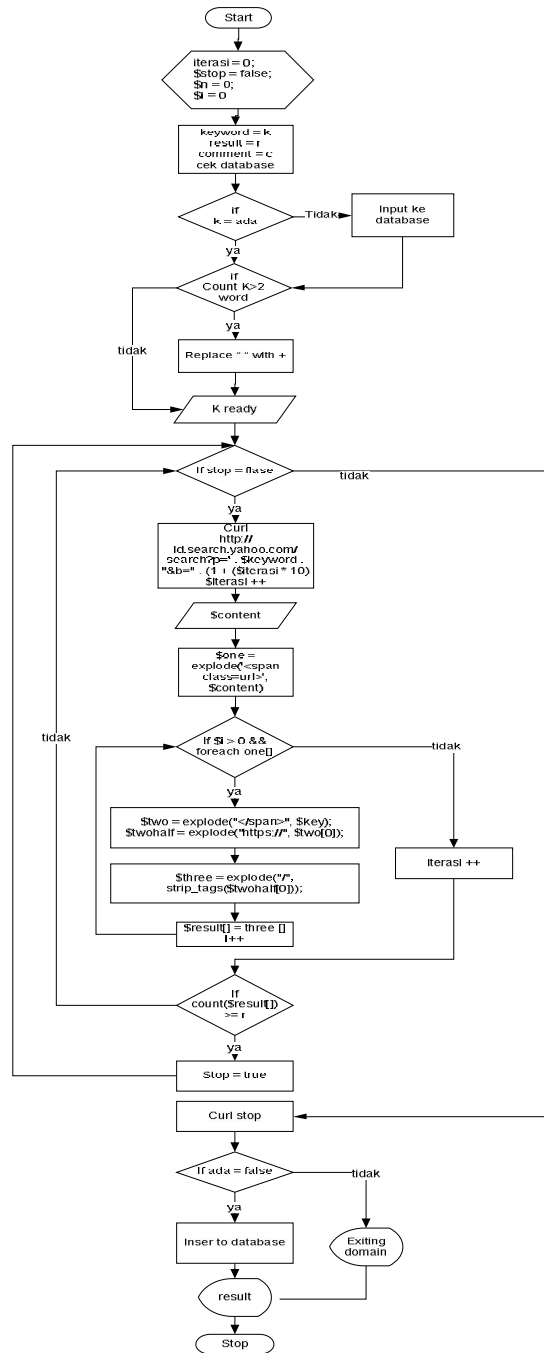


The image shows a wireframe of a web page for 'Proxy Setting Squidsearch'. At the top, there is a header area with the word 'LOGO' centered. Below the header is a navigation bar with links: 'Uri Regex', 'Keyword Search', 'Blocked List', 'Proxy Setting' (highlighted in red), 'Manage Userpanel', and 'Logout'. The main content area is enclosed in a rounded rectangle and contains two sections. The first section is titled 'Action Status' and contains four buttons: 'Start', 'Stop', 'Restart', and 'Reconfigure'. The second section is titled 'Auto proxy' and contains a single text input field. At the bottom of the main content area, there is a footer that reads 'copyright squidsearch.com'.

**Gambar 3.23** Rancangan Halaman *Proxy Setting Squidsearch*

### 3.6 Metode Ekstraksi Hasil Pencarian *Search Engine*

Gambar 3.24 merupakan proses ekstraksi hasil pencarian dari *search engine*. Digambarkan dengan proses berbentuk *flowchart*.



Gambar 3.24 Flowchart hasil pencarian *search engine*

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Implementasi Secara Umum**

Implementasi merupakan tahap akhir dari sebuah sistem, apakah sistem dapat berjalan dengan baik atau tidak, dengan memberikan keluaran dari sistem yang telah direncanakan sebelumnya, dan bagaimana sistem dapat sesuai dengan hasil rancangan pada BAB III.

#### **4.2 Tahapan Implementasi Perangkat Lunak**

Dalam pengaplikasian perangkat lunak ini, terdapat beberapa tahap yang terdiri dari :

1. Instalasi dan Konfigurasi *Proxy*  
Pada tahap ini dilakukan instalasi *Proxy Server* dengan menggunakan aplikasi Squid 3.0.
2. Instalasi dan Konfigurasi Basisdata Mysql  
Pada tahap ini dilakukan instalasi basisdata MySQL versi 5.1.37 yang digunakan untuk menyimpan basisdata sistem *squidsearch*.
3. Instalasi dan *PhpMyadmin*  
Pada tahap ini dilakukan instalasi terhadap *phpmyadmin* yang nantinya akan digunakan untuk mempermudah manajemen basisdata mysql.
4. *Instalasi cron*  
Pada tahap ini dilakukan instalasi terhadap aplikasi cron, yang nantinya akan digunakan sebagai alat *otomatic update blocked list*.
5. Instalasi dan Konfigurasi *Web Server Apache2*  
Pada tahap ini dilakukan instalasi *web Server apache2* yang akan digunakan sebagai tempat untuk menyimpan sistem *squidsearch* yang berbasis web.

#### 6. Instalasi Modul Php5-Curl

Pada tahap ini dilakukan instalasi modul *php5-curl* yang digunakan sebagai tambahan modul yang dibutuhkan sistem *squidsearch* untuk dapat mengekstrak hasil pencarian yang berasal dari *search engine*.

#### 7. Pembuatan *Shell Script*

Pada tahap ini dilakukan pembuatan *script* menggunakan bahasa pemrograman *shell* yang berfungsi untuk menambah fitur *Server* yaitu fungsi bloking dan konfigurasi daemon *squid*.

#### 8. Pembuatan *Interface Web* untuk Konfigurasi *Squidsearch*

Pada tahap ini dilakukan pembuatan *web* yang berfungsi untuk mempermudah pengaturan pada *server*.

### 4.3 Implementasi Tahap Perancangan

#### 4.3.1 Instalasi dan konfigurasi *proxy*

1. Pada tahap ini dibutuhkan paket *source code* dari *squid* yang didapatkan dari *website* <http://www.squid-cache.org/Versions/> secara gratis.

2. Instalasi paket dengan ekstraksi paket *source code*

```
$ tar -xzf squid-3.0.STABLE25.tar.gz
$ cd squid-3.0.STABLE25/
```

3. Kompilasi hasil ekstraksi

```
$ ./configure --prefix=/usr/local/squid
$ make
$ sudo make install
```

4. Ubah kepemilikan *squid*

```
##pengubahan kepemilikan dan hak akses terhadap direktori.
#chown Squidsearch-Server: Squidsearch-Server
/usr/local/squid/
#chmod 755 /usr/local/squid/
#cd /usr/local/squid/bin
#chown Squidsearch-Server: Squidsearch-Server. chown
Squidsearch-Server: Squidsearch-Server *
```

```

#chmod 755 . *
#cd /usr/local/squid/sbin
#chown Squidsearch-Server: Squidsearch-Server. chown
Squidsearch-Server: Squidsearch-Server *
#chmod 755 . *
#cd /usr/local/squid/etc
#chmod 2775 .
#chown Squidsearch-Server: Squidsearch-Server. *
#chown squid:squidadm /usr/local/squid/var/logs
#chmod 770 /usr/local/squid/var/logs
###pembuatan dan perubahan kepemilikan serta hak akses.
#mkdir /usr/local/squid/var/cache/
#chown Squidsearch-Server: Squidsearch-Server
/usr/local/squid/var/cache/
#chmod 770 /usr/local/squid/var/cache/

```

## 5. Konfigurasi squid.conf:

```

#Configuration for authentication
auth_param basic children 5
auth_param basic realm Squid proxy-caching web server
auth_param basic credentialsttl 2 hours
#define acl by port and network
acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32
acl localnet src 10.0.0.0/8
acl localnet src 172.16.0.0/12
acl localnet src 192.168.0.0/16a
acl SSL_ports port 443
acl Safe_ports port 80 # http
acl Safe_ports port 21 # FTP
acl Safe_ports port 443 # https
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl CONNECT method CONNECT

```

```
#define access list and http_acces
#http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
#http_access allow localnet
#=====rules=====#
#http_access allow localnet
#http_access allow localhost
#And finally deny all other access to this proxy
acl url_regex url_regex -i "/usr/local/squid/acl/url_regex"
acl blocked_list dstdomain -i "/usr/local/squid/acl/blocked_list"
http_access deny url_regex
http_access deny blocked_list
http_access allow all
#squid pord uses by client
http_port 3128
#memory and replacement algoritm
cache_mem 64 MB
memory_replacement_policy lru
cache_replacement_policy lru
cache_dir ufs /usr/local/squid/var/cache 100 16 256
#define maximum and minimum object size
minimum_object_size 0 KB
maximum_object_size 4096 KB
#define file log
access_log /usr/local/squid/var/logs/access.log
cache_store_log /usr/local/squid/var/logs/store.log
#define user and group squid
visible_hostname squidsearch-server
log_icp_queries on
#define error handling file
error_directory /usr/local/squid/share/errors/templates
```



## 6. Buat *cache* squid

```
$/usr/lokal/squid/sbin/squid -z
```

## 7. Start squid

```
$/usr/lokal/squid/sbin/squid start
$/usr/lokal/squid/sbin/squid -k reconfigure
```

### 4.3.2 Instalasi dan konfigurasi basisdata MySQL

1. Pada tahap ini dilakukan instalasi perangkat lunak MySQL sebagai *Server* basis data. Jenis paket yang digunakan bertipe *binary*, diinstal dengan perintah `apt-get`

```
$ sudo apt-get install mysql-Server-5.1
```

### 4.3.3 Instalasi phpmyadmin

1. Pada tahap ini dilakukan instalasi perangkat lunak MySQL sebagai *Server* basis data. Jenis paket yang digunakan bertipe *binary*, diinstal dengan perintah `apt-get`

```
$ sudo apt-get install phpmyadmin
```

2. Jalankan *script* untuk membuat tabel

```
CREATE TABLE IF NOT EXISTS `admin` (
  `id_admin` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  PRIMARY KEY (`id_admin`),
  UNIQUE KEY `id_admin` (`id_admin`),
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1
;
INSERT INTO `admin` (`id_admin`, `username`, `password`)
VALUES
(1, 'admin', 'bf6006a821b6b7fe6f26e9b9054f4b2d');
CREATE TABLE IF NOT EXISTS `cron` (
  `cron_id` int(10) NOT NULL AUTO_INCREMENT,
  `m` int(3) NOT NULL,
  `h` int(3) NOT NULL,
  `dom` int(3) NOT NULL,
  `mon` int(3) NOT NULL,
  `user` varchar(100) NOT NULL,
  `command` varchar(255) NOT NULL,
```

```

PRIMARY KEY (`cron_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
CREATE TABLE IF NOT EXISTS `keyword` (
  `id_keyword` int(11) NOT NULL AUTO_INCREMENT,
  `keyword` varchar(255) NOT NULL,
  `result` int(11) NOT NULL,
  `comment` varchar(255) NOT NULL,
  PRIMARY KEY (`id_keyword`),
  UNIQUE KEY `id_keyword` (`id_keyword`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
INSERT INTO `keyword` (`id_keyword`, `keyword`, `result`,
`comment`) VALUES
(1, 'blockedlist', 0, 'default for blocked list'),
(2, 'white list', 0, 'default keyword for white list');
CREATE TABLE IF NOT EXISTS `url_regex` (
  `id_url_regex` int(255) NOT NULL AUTO_INCREMENT,
  `regex` varchar(255) NOT NULL,
  `comment` varchar(255) NOT NULL,
  PRIMARY KEY (`id_url_regex`),
  UNIQUE KEY `id_url_regex` (`id_url_regex`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
INSERT INTO `url_regex` (`id_url_regex`, `regex`,
`comment`) VALUES
(1, 'sex', 'blok website with sex');
CREATE TABLE IF NOT EXISTS `userpanel` (
  `id_userpanel` int(255) NOT NULL AUTO_INCREMENT,
  `realname` varchar(255) NOT NULL,
  `username` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  PRIMARY KEY (`id_userpanel`),
  UNIQUE KEY `username` (`username`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
INSERT INTO `userpanel` (`id_userpanel`, `realname`,
`username`, `password`) VALUES
(1, 'NOC 1', 'userpanel',
'2fff96ddee09f2221bcefe34c68b324e');
CREATE TABLE IF NOT EXISTS `website_list` (
  `id_site` int(11) NOT NULL AUTO_INCREMENT,
  `url_list` text NOT NULL,
  `status` int(1) NOT NULL,
  `id_keyword` int(11) NOT NULL,
  `comment` varchar(255) NOT NULL,
  PRIMARY KEY (`id_list`),
  UNIQUE KEY `id_list` (`id_list`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

#### 4.3.4 Instalasi dan konfigurasi *Web Server Apache2*

1. Pada tahap ini dilakukan instalasi perangkat lunak apache2 sebagai *web Server*. Jenis paket yang digunakan bertipe *binary*, diinstal dengan perintah apt-get

```
$ sudo apt-get install php5 apache2 php5-curl
```

#### 4.3.5 Instalasi modul *php5-curl*

```
$ sudo apt-get install php5 apache2 php5-curl
```

#### 4.3.6 Instalasi *crontab*

```
$ sudo apt-get install cron
```

#### 4.3.7 Pembuatan *shell script*

1. Pada tahap ini dilakukan pembuatan *script* yang digunakan oleh sistem untuk menjalankan fungsinya
2. Pembuatan folder tempat menyimpan *script* dan *acl* dari sistem *squid*

```
mkdir /usr/lokal/squid/script
mkdir /usr/lokal/squid/acl
```

3. Pembuatan *script* dan *acl* di folder yang sudah dibuat

```
touch /usr/lokal/squid/script/export_blocked_list.sh
touch /usr/lokal/squid/script/export_url_regex.sh
touch /usr/lokal/squid/script/reconfsqid.sh
touch /usr/lokal/squid/script/restartsqid.sh
touch /usr/lokal/squid/script/startsqid.sh
touch /usr/lokal/squid/script/statussqid.sh
touch /usr/lokal/squid/script/startsqid/stopsquid.sh
touch /usr/lokal/squid/acl/blocked_list
touch /usr/lokal/squid/acl/blocked_list_temp
touch /usr/lokal/squid/acl/url_regex
touch /usr/lokal/squid/acl/url_regex_temp
```

- export\_blocked\_list.sh

```
mysql -u root -p Squidsearch --password="root" --
execute="SELECT url_list FROM website_list where status=0
ORDER BY url_list" > /usr/lokal/squid/acl/blocked_list_temp
tail -n +2 /usr/lokal/squid/acl/blocked_list_temp >
/usr/lokal/squid/acl/blocked_list
/usr/lokal/squid/sbin/squid -k reconfigure
```

- export\_url\_regex.sh

```
mysql -u root -p Squidsearch --password="root" --
execute="SELECT regex FROM url_regex ORDER BY regex" >
/usr/lokal/squid/acl/url_regex_temp
tail -n +2 /usr/lokal/squid/acl/url_regex_temp >
/usr/lokal/squid/acl/url_regex
```

- reconfsqid.sh

```
/usr/lokal/squid/sbin/squid -k reconfigure
```

- restartsqid.sh

```
/usr/lokal/squid/sbin/squid restart
```

- startsqid.sh

```
/usr/lokal/squid/sbin/squid start
```

- statussqid.sh

```
/usr/lokal/squid/sbin/squid status
```

- stopsqid.sh

```
/usr/lokal/squid/sbin/squid stop
```

- blocked\_list

Merupakan tempat penyimpanan file basisdata *proxy* untuk *blocked list*.

- blocked\_list\_temp

Merupakan tempat penyimpanan file basisdata *proxy* untuk *blocked list* sementara.

- url\_regex

Merupakan tempat penyimpanan file basisdata *proxy* untuk *url\_regex*.

- `url_regex_temp`

Merupakan tempat penyimpanan file basisdata regex untuk *url\_regex* sementara.

#### 4.3.8 Pembuatan *script interface* untuk konfigurasi *Server*

Pada tahap ini dilakukan pembuatan *web* yang berfungsi untuk mempermudah pengaturan pada *Server*

```
<?php
session_start();
include "database_con.php";
function Only1br($string) {
    return preg_replace("/(\r\n)+|(\n|\r)+/", "", $string);
}
if ($_POST["result"] != "") {
    $katakunci = $_POST["keyword"];
    $menu = mysql_query("select * from keyword WHERE keyword
= '" . $katakunci . "'");
    $ada = false;
    while ($data = mysql_fetch_array($menu)) {
        $ada = true;
        $_SESSION["keywordAda"] = true;
    }
    if ($ada == false) {
        $sql = "INSERT INTO keyword(keyword,comment,result)
VALUES ('" . $katakunci . "', '" . $_POST["comment"] . "', '" .
$_POST["result"] . "')";
        mysql_query($sql);
    }
    $_SESSION["keyword"] = $katakunci;
    $menu = mysql_query("select * from keyword WHERE keyword
= '" . $katakunci . "'");
    while ($data = mysql_fetch_array($menu)) {
        $id = $data["id_keyword"];
    }
}
```

```

}
$search = array(' ');
$replace = array('+');
$hasil = str_replace($search, $replace,
$_POST["keyword"]);
$iterasi = 0;
$ch = curl_init();
$stop = false;
$n = 0;
set_time_limit(0);
while ($stop == false) {
    curl_setopt($ch, CURLOPT_COOKIEJAR, 'cookie.txt');
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    $store = curl_exec($ch);
    curl_setopt($ch, CURLOPT_URL,
'http://id.search.yahoo.com/search?p=' . $hasil . "&b=" . (1
+ ($iterasi * 10)));
    $content = curl_exec($ch);
    $one = explode('<span class=url>', $content);
    $i = 0;
    foreach ($one as $key) {
        if ($i > 0) {
            $two = explode("</span>", $key);
            $twohalf = explode("https://", $two[0]);
            $j = 0;
            foreach ($twohalf as $key2) {
                $j++;
            }
            $three = explode("/",
strip_tags($twohalf[0]));
            if ($three[0] == "") {
                $three = explode("/",
strip_tags($twohalf[1]));
            }
        }
    }
}

```

```

        if (count($result) > 0) {
            if (!in_array($three[0], $result)) {
                $result[] = $three[0];
                // echo "<li>" . $three[0] . "</li>";
                $n++;
                if ($n >= $jumlah) {
                    $stop = true;
                    break;
                }
            }
        }
        if (count($result) == 0) {
            $result[] = $three[0];
            // echo "<li>" . $three[0] . "</li>";
            $n++;
            if ($n >= $jumlah) {
                $stop = true;
                break;
            }
        }
    }
    $i++;
}
$iterasi++;
}
curl_close($ch);
foreach ($result as $key) {
    $menu = mysql_query("select * from website_list WHERE
url_list = '$key'");
    $sada = false;
    while ($data = mysql_fetch_array($menu)) {
        $sada = true;
    }
    if (!$sada) {
        //echo "<li>" . $key . "</li>";
    }
}

```

```
    $sql = "INSERT INTO
website_list(url_list,comment,status,id_keyword) VALUES ('" .
$key . "','" . $key . "','0','" . $id . "')";
    mysql_query($sql);
    $_SESSION["list"][] = "<b>" . $key . "</b>";
} else {
    // echo "<li>" . $key . " - resu</li>";
    $_SESSION["list"][] = "<b>" . $key . "</b> -
<small>existing domain</small>";
}
}
}
shell_exec('sh
/usr/local/squid/script/export_blocked_list.sh');
    shell_exec('sh
/usr/local/squid/script/reconfsqid.sh');
?>
<script>
    window.location="../manage_keyword_search.php";
</script>
```



### 3.4 Hasil Perancangan

#### 3.4.1 Halaman *Login Squidsearch*

**Gambar 4.1** Halaman *Login Squidsearch*

Gambar 4.1 merupakan tampilan awal dari sistem pada saat pertama kali diakses. Diawali dengan halaman *Login*. *User* memasukkan *username* dan *password* untuk masuk ke dalam sistem, *user* juga memasukkan tipe akses yang diinginkan, baik *userpanel* ataupun *admin*.

#### 3.4.2 Halaman *Manage Url Regex*

NO	REGEX	COMMENT	DELETE/EDIT
1	tube	blok website with tube	✖ 📄
2	drugs	blok website with drugs	✖ 📄
3	porn	blok website with porn	✖ 📄
4	sex	blok website with sex	✖ 📄

**Gambar 4.2** Halaman *Manage Url Regex*

Gambar 4.2 merupakan halaman *Url regex*. Di halaman ini *userpanel* dapat memasukkan suatu *regex* ke dalam sistem *squidsearch*, yang nantinya akan digunakan sebagai acuan sistem *proxy* untuk melarang koneksi klien ke *website* yang mempunyai url kondisi sesuai dengan aturan *regex*. *User*

juga dapat menghapus *regex* yang ada ataupun mengedit *regex* yang sudah dimasukan kedalam sistem.

### 3.4.3 Halaman *Add Url Regex*

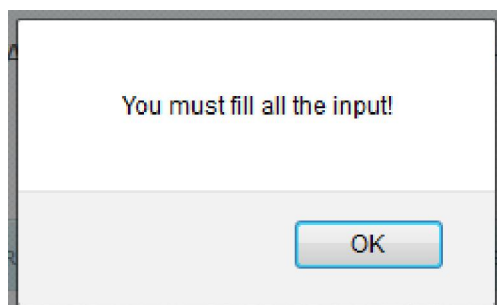
REGEX	COMMENT	ACTION
girl	blok website with girl	Save

NO	REGEX	COMMENT	DELETED/EDIT
1	tube	blok website with tube	✕ ✎
2	drugs	blok website with drugs	✕ ✎
3	porn	blok website with porn	✕ ✎
4	sex	blok website with sex	✕ ✎

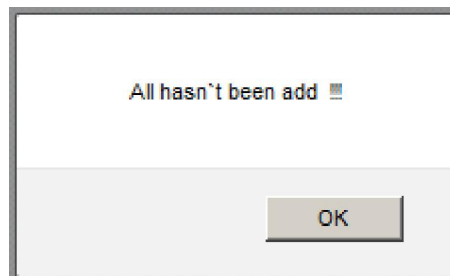
**Gambar 4.3** Halaman *Add Url Regex*

Gambar 4.3 merupakan halaman *add url regex*, di halaman ini *user* dapat menambahkan *regex* yang ada di sistem *squidsearch*. Pada halaman ini user diwajibkan mengisi data berupa *regex* yang diinginkan maupun *comment*-nya, *coment* disini merupakan definisi dari fungsi *regex* yang diinputkan.



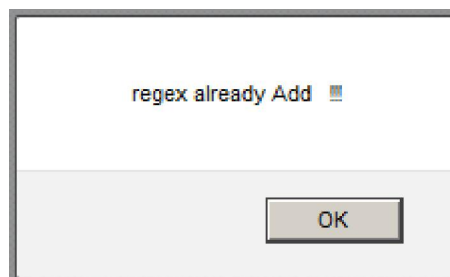
**Gambar 4.4** Konfirmasi *Url Regex*

Gambar 4.4 merupakan konfirmasi dari sistem bahwa proses *Add* telah berhasil dilakukan.



**Gambar 4.5** Laporan Data Tidak Lengkap

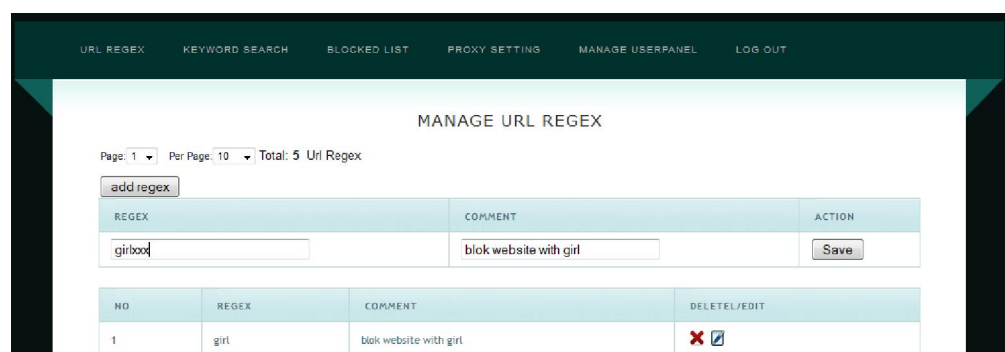
Gambar 4.5 Merupakan laporan bahwa data yang dimasukkan belum lengkap.



**Gambar 4.6** Laporan *Regex* Sudah Dimasukan

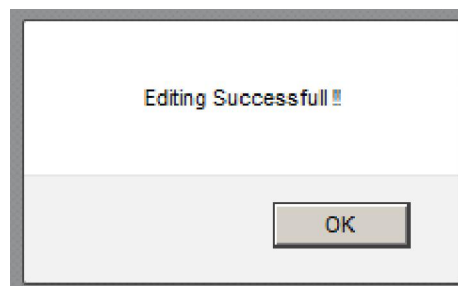
Gambar 4.6 Merupakan laporan bahwa *regex* yang dimasukkan sudah ada.

### 3.4.4 Halaman *Edit Url Regex*



**Gambar 4.7** Halaman *Edit Url Regex*

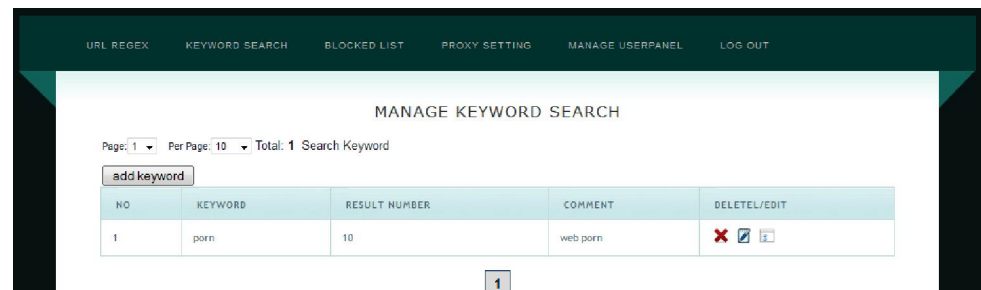
Gambar 4.7 merupakan halaman edit dimana *user* dapat melakukan *editing* terhadap *url regex* yang telah dimasukan kedalam sistem *squidsearch*.



**Gambar 4.8** Konfirmasi editing succesfull

Gambar 4.8 merupakan konfirmasi *editing successful* gambar dari sistem bahwa proses *editing* telah berhasil dilakukan.

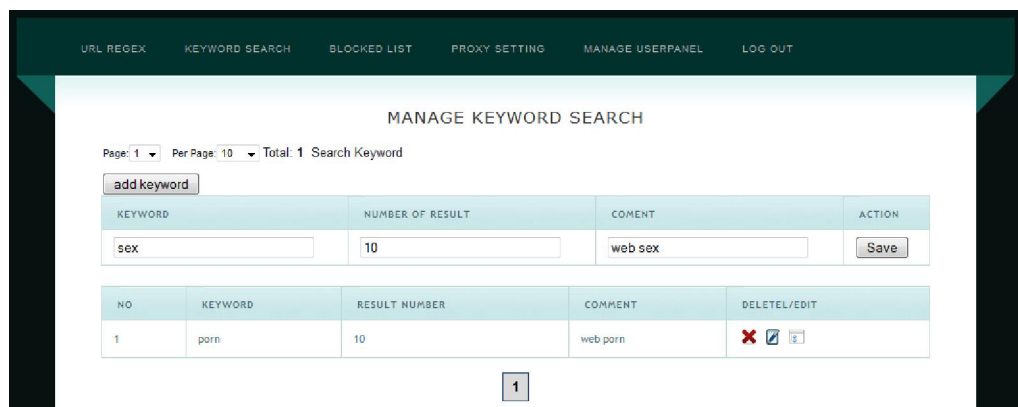
### 3.4.5 Halaman *Manage Keyword Search*



**Gambar 4.9** Halaman *Manage Keyword Search*

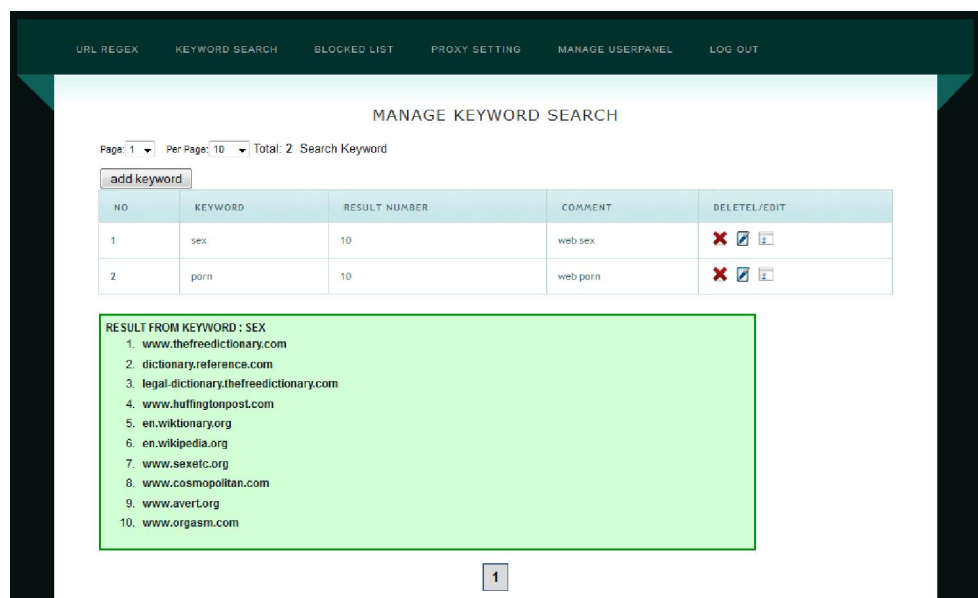
Gambar 4.9 Merupakan halaman *manage Keyword Search* dimana *userpanel* dapat mengatur *keyword* apa saja yang dapat digunakan sebagai acuan sistem untuk meminta *query* ke *search engine*. Di sini user dapat memasukan *keyword* dan berapa jumlah yang diminta ke *search engine* Pada halaman ini *user* juga dapat melakukan *edit delete* ataupun *Add* pada sistem *squidsearch*.

### 3.4.6 Halaman *Add Keyword Search*



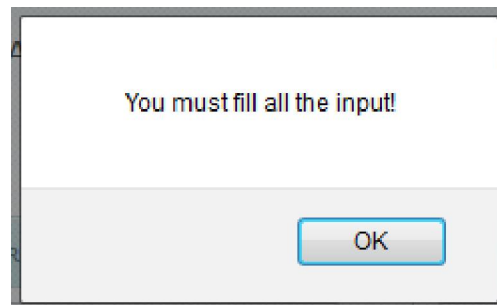
**Gambar 4.10** Halaman *Add Keyword Search*

Gambar 4.10 merupakan halaman *Add Keyword Search*, pada halaman ini kita diminta memasukan *keyword* yang di inginkan pada form, jumlah *result* yang diminta dan *comment*.



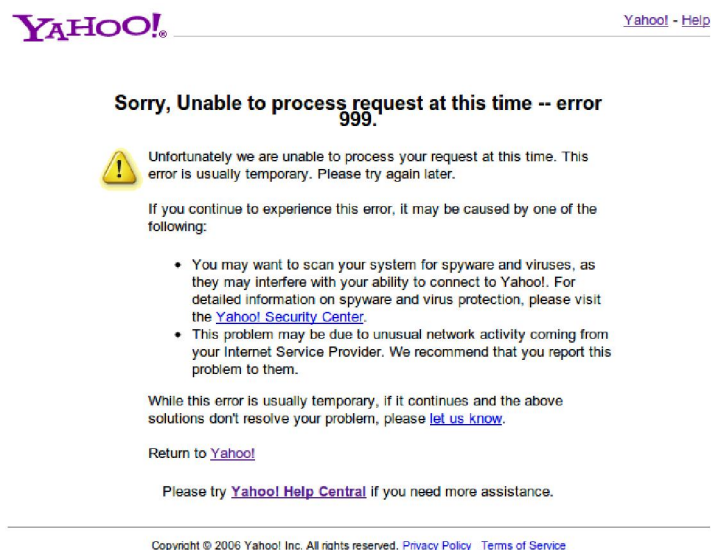
**Gambar 4.11** Hasil Pencarian *Keyword Search*

Gambar 4.11 merupakan konfirmasi yang diberikan oleh sistem *squidsearch* saat *user* melakukan masukan terhadap *search key*, yang nantinya akan digunakan sebagai *keyword* oleh *search engine*.



**Gambar 4.12** Laporan Data Tidak Lengkap

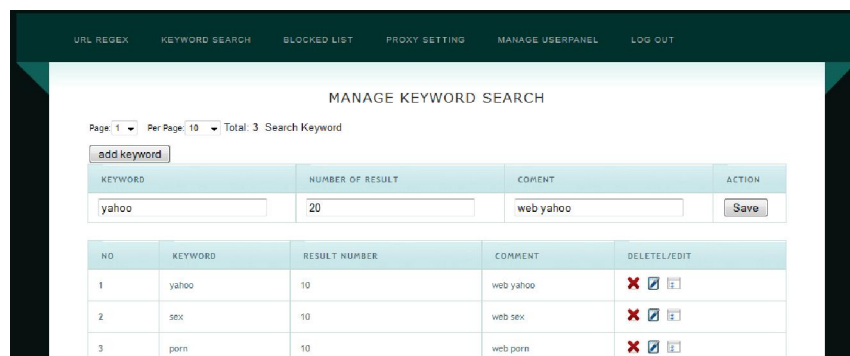
Gambar 4.12 Merupakan laporan bahwa data yang dimasukan belum lengkap.



**Gambar 4.13** Gambar Laporan *Baned process request Error*

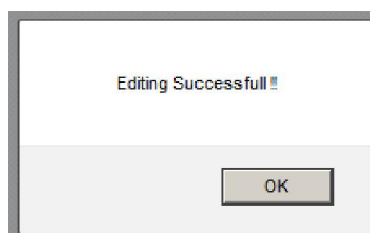
Gambar 4.13 merupakan tampilan dari sistem *squidsearch* apabila *search engine* melarang request *squidsearch* yang mencurigakan.

### 3.4.7 Halaman *Edit Keyword Search*



**Gambar 4.14** Halaman *Edit Keyword Search*

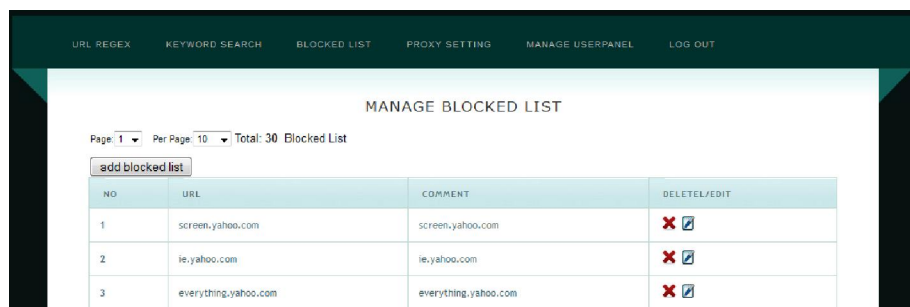
Gambar 4.14 merupakan halaman *edit keyword search*. merupakan halaman yang digunakan untuk dapat mengedit *keyword* dan jumlah hasil pencarian yang akan diminta ke *search engine*.



**Gambar 4.15** Konfirmasi *editing successful*

Gambar 4.15 merupakan konfirmasi dari sistem bahwa proses *editing* telah berhasil dilakukan.

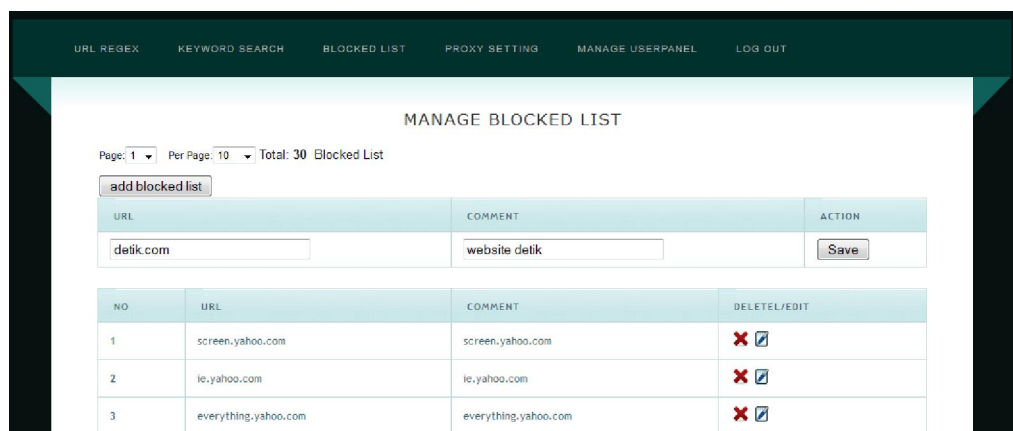
### 3.4.8 Halaman *Manage Blocked List*



**Gambar 4.16** Halaman *manage blocked list*

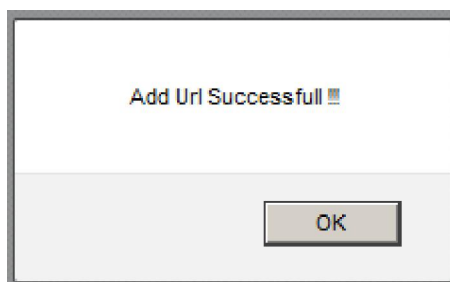
Gambar 4.16 merupakan halaman manajemen *blocked list*, merupakan halaman utama dari *manage blocked site* yang digunakan sebagai manajemen *website* apa saja yang dilarang untuk diakses oleh klien. *Userpanel* dapat melakukan *editing Add* ataupun *delete* terhadap list *website* yang ada didalam sistem.

### 3.4.9 Halaman *Add Blocked List*



**Gambar 4.17** Halaman *Add Blocked List*

Gambar 4.17 merupakan halaman *add blocked list*, merupakan halaman yang menunjukkan *website* atau *domain* apa saja yang dilarang diakses oleh klien.

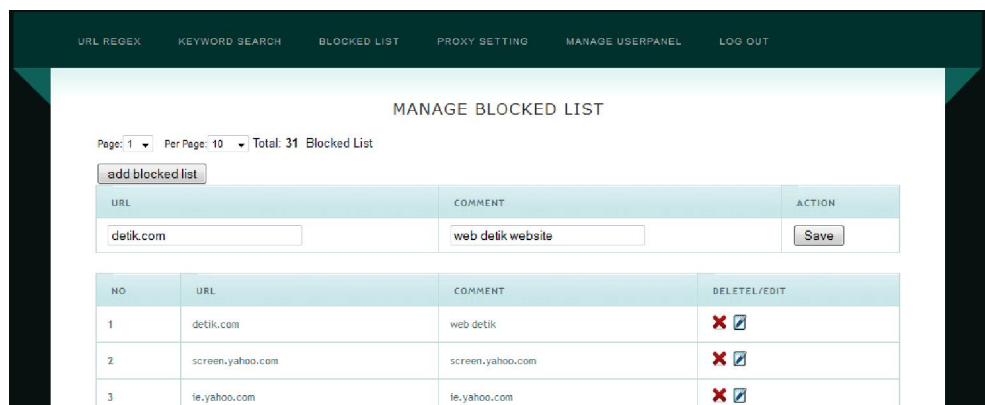


**Gambar 4.18** Konfirmasi *Add Url Successful*

Gambar 4.18 merupakan konfirmasi yang diberikan dari sistem bahwa url sudah masuk kedalam *blocklist*.

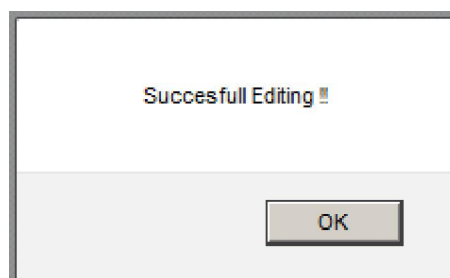


### 3.4.10 Halaman *Edit Blocked List*



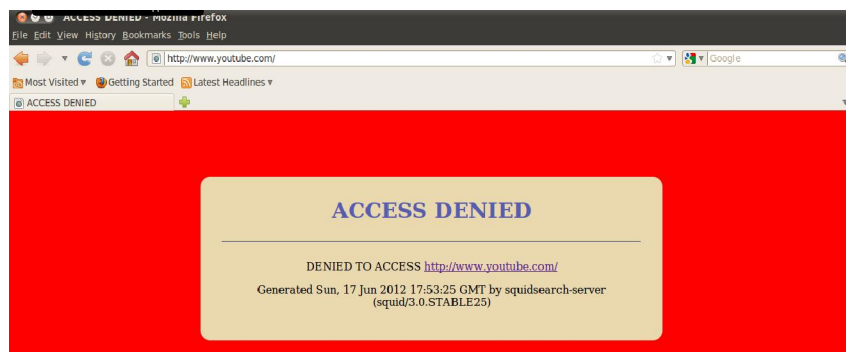
**Gambar 4.19** Halaman *Edit Blocked List*

Gambar 4.19 merupakan halaman edit blocked list, pada halaman ini admin dapat melakukan perubahan terhadap daftar domain yang akan di blok oleh proxy.



**Gambar 4.20** Konfirmasi *Successful Editing*

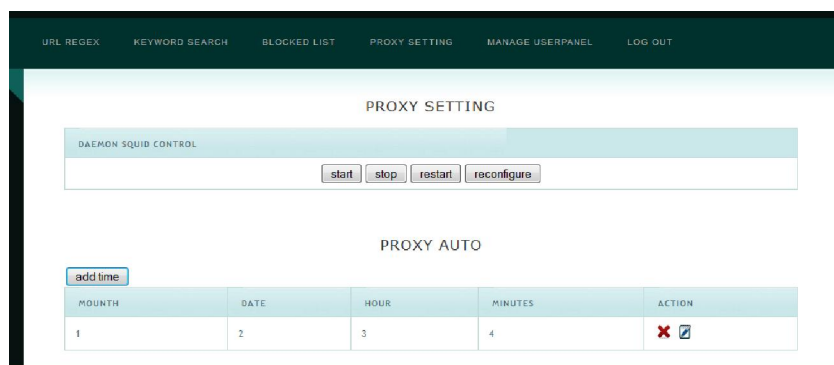
Gambar 4.20 merupakan konfirmasi dari sistem bahwa proses *editing* telah berhasil dilakukan.



**Gambar 4.21** Halaman *Access Denied*

Gambar 4.21 merupakan tampilan halaman dari klient yang mengakses situs yang di blok.

### 3.4.11 Halaman *Proxy Setting*



**Gambar 4.22** Halaman *proxy Setting*

Gambar 4.22 merupakan halaman yang mengatur *daemon squid* pada halaman ini user dapat melakukan *stop*, *start*, *restart* ataupun *reconfigure* pada *daemon squid*, disini juga sistem dapat melakukan penjadwalan kapan proses *update* ke *search engine* dapat dilakukan.



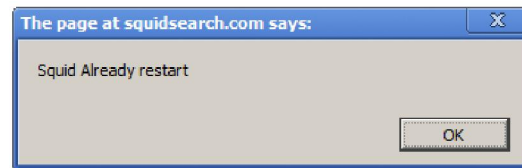
**Gambar 4.23** Konfirmasi *squid start*

Gambar 4.23 merupakan konfirmasi bahwa proses perestaran *squid* telah dilakukan.



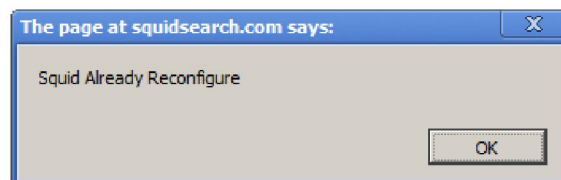
**Gambar 4.24** Konfirmasi *squid stop*

Gambar 4.24 merupakan konfirmasi bahwa proses pembunuhan *squid* telah dilakukan.



**Gambar 4.25** Konfirmasi *squid restart*

Gambar 4.25 merupakan konfirmasi bahwa proses perestaran *squid* telah dilakukan.



**Gambar 4.26** Konfirmasi squid reconfigure

Gambar 4.26 merupakan konfirmasi bahwa proses reconfigure *squid* telah dilakukan

add time

MONTH	DATE	HOUR	MINUTES	ACTION
<input type="text" value="12"/>	<input type="text" value="12"/>	<input type="text" value="12"/>	<input type="text" value="12"/>	<input type="button" value="Save"/>

MONTH	DATE	HOUR	MINUTES	ACTION
1	2	3	4	<input type="button" value="✕"/> <input type="button" value="✎"/>

**Gambar 4.27** Halaman *Automatic Update Squid*

Gambar 4.27 merupakan gambar halaman *otomatic update squid*. Dimana admin dapat menentukan waktu yang digunakan oleh sistem *squid search* dalam melakukan perubahan basisdatanya.

### 3.4.12 Halaman Manajemen *Userpanel*

**SQUIDSEARCH**  
SQUID SYSTEM + SEARCH ENGINE

URL REGEX   KEYWORD SEARCH   BLOCKED LIST   PROXY SETTING   MANAGE USERPANEL   LOG OUT

MANAGE USERPANEL

Page:  Per Page:  Total: 5 Userpanel

NO	REALNAME	USERNAME	DELETE/EDIT
1	NOC 1	userpanel1	<input type="button" value="✕"/> <input type="button" value="✎"/>
2	NOC 2	userpanel2	<input type="button" value="✕"/> <input type="button" value="✎"/>
3	NOC 3	userpanel3	<input type="button" value="✕"/> <input type="button" value="✎"/>
4	NOC 4	userpanel4	<input type="button" value="✕"/> <input type="button" value="✎"/>
5	NOC 5	userpanel5	<input type="button" value="✕"/> <input type="button" value="✎"/>

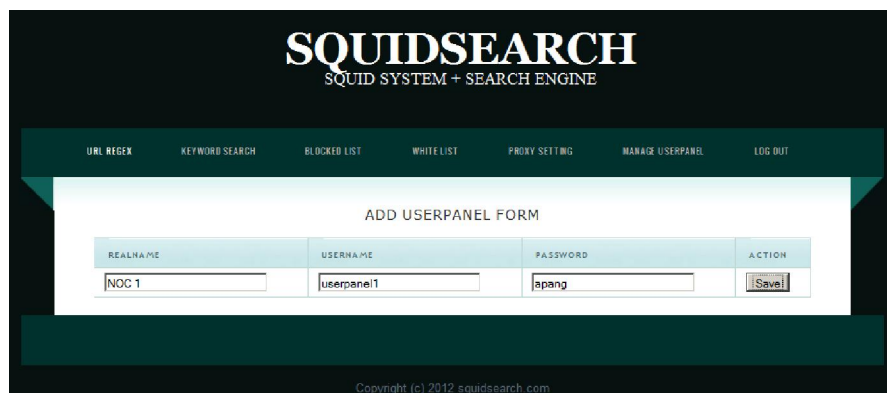
1

Copyright (c) 2012 squidsearch.com

**Gambar 4.28** Halaman manajemen *userpanel*

Gambar 4.28 merupakan halaman dimana admin dapat memajemen *userpanel* yang notabene adalah pengguna sistem, admin dapat menghapus menambah ataupun mengubah informasi tentang *userpanel* yang ada pada sistem

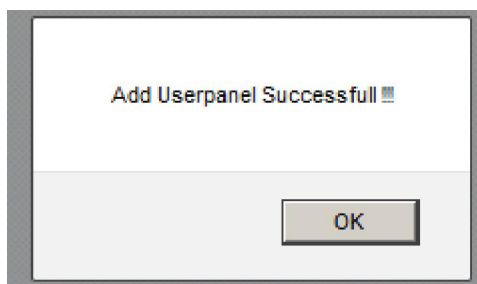
### 3.4.13 Halaman *Add Userpanel*



REALNAME	USERNAME	PASSWORD	ACTION
NOC 1	userpanel1	lapang	Save!

**Gambar 4.29** Halaman *Add userpanel*

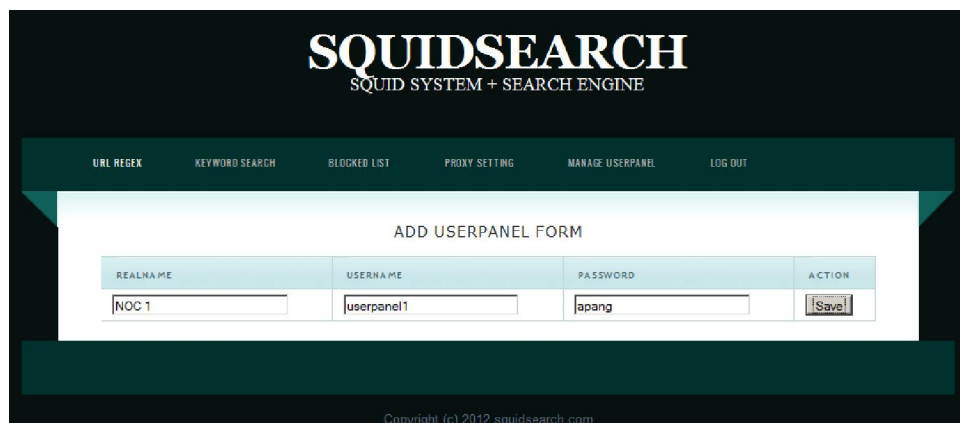
Gambar 4.29 merupakan halaman dimana admin memasukan informasi untuk *userpanel* yang akan menggunakan *squid* admin wajib mengisi *realname* *username* *password* .



**Gambar 4.30** Konfirmasi *Add userpanel* successful

Gambar 4.30 merupakan konfirmasi yang diberikan oleh sistem apabila sudah memasukan *userpanel* yang akan menggunakan sistem.

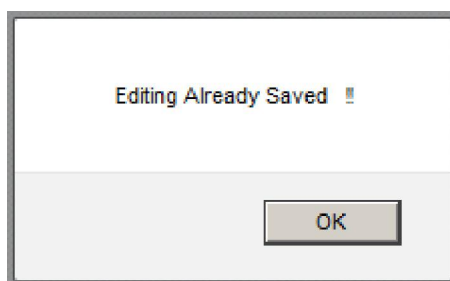
### 3.4.14 Halaman *Edit Userpanel*



REALNAME	USERNAME	PASSWORD	ACTION
NOC 1	userpanel1	japang	Save!

**Gambar 4.31** Halaman *edit userpanel*

Gambar 4.31 Merupakan halaman edit user panel dimana admin bisa melakukan *editing* terhadap *userpanel* pengguna sistem dimulai dengan memasukan *username* yang ingin dirubah, ataupun *password* apabila diinginkan.



**Gambar 4.32** Konfirmasi *editing successfull*

Gambar 4.32 Merupakan konfirmasi yang diberikan oleh sistem pada saat sistem sukses melakukan *editing* terhadap *userpanel*.

## BAB V PENUTUP

### 5.1 Kesimpulan

Sistem *Automatic Update Web Filtering* menggunakan *Squid* dan *Search Engine* Telah selesai dibangun dan diupayakan sesuai dengan perencanaan dan perancangan pada tahap analisis kebutuhan perangkat keras, perangkat lunak, desain dan alur data aplikasi serta desain jaringan implementasi. Setelah melakukan tahap pengujian, maka dapat diambil kesimpulan bahwa:

1. Sistem *Automatic update web Filtering squid + search engine* dapat berjalan.
2. Antarmuka sistem yang berbasis *web* mempermudah *Administrator* untuk dapat mengontrol akses klien ke *internet*.
3. Fungsi *Automatic update* yang ada pada sistem bekerja akan tetapi ada batasan query terhadap *search engine* tergantung pada batasan yang diberikan oleh *search engine*.

### 5.2 Saran

Terdapat beberapa hal yang perlu diperhatikan ketika kita akan mengimplementasikan sistem ini. Supaya langkah-langkah bisa berjalan dengan baik, ada beberapa saran yang perlu diperhatikan, yaitu pada penelitian selanjutnya, disarankan :

1. Pengembangan fungsi pada sistem *squidsearch* dengan jumlah *query* yang lebih banyak, dengan menggunakan service yang disediakan dari pihak *search engine*
2. Perlu ditambahkan menu-menu yang lebih menyeluruh mengenai manajemen squid bukan hanya kolaborasi dengan *search engine*, akan tetapi keseluruhan konfigurasi proxy.

3. Perlu dilakukan penelitian lebih lanjut bagaimana mendapatkan hasil pencarian yang paling optimal pada hasil pencarian *search engine* sesuai kebutuhan sistem *proxy*.



**DAFTAR PUSTAKA**

- (Heather,2005:1) Wallace, Heather. 2005. *Firewall* Protection Burning Bridges is Bad, But *Firewalls* are Good (On-line). Available at <http://www.updatexp.com>
- (John,2002:3) John Wack, Ken Cutler, Jamie Pole. 2002. Guidelines on *Firewalls* and *Firewall* policy (On Line) Available at [scholar.search engine.co.id](http://scholar.searchengine.co.id)
- (Sirkel:2008,17)
- (Santoso, 2012:24) Santoso, Hari. ?. ? (On-line). Available at [lib.uin-malang.ac.id/thesis/chapter\\_ii/06550023-hari-santoso.ps](http://lib.uin-malang.ac.id/thesis/chapter_ii/06550023-hari-santoso.ps) (diakses tanggal 4 mei 2012)
- (Nugroho,2011:13) Nugroho, Lantip Agil. 2011. Remastering Distro Ubuntu untuk *Gateway*, *Distributed Filtering* dan *Security*. Skripsi, tidak diterbitkan. Yogyakarta: Fakultas Teknologi Industri Universitas Islam Indonesia.





SARAN/USULAN PRESENTASI KEMAJUAN TUGAS AKHIR

Nama Mhs. : Citra Arfanain.  
No. Mhs. : 07-261  
Judul TA : \_\_\_\_\_

o. sering komentar  
o. buat timeline biar tak wabor!

Nilai kemajuan Tugas Akhir: \_\_\_\_\_ (0 - 100)  
(studi pustaka, perancangan, penguasaan materi, ketepatan)

Yogyakarta, 20/9/17.....

Dosen,

U. Sora

ISNANI, F.  
(nama terang)