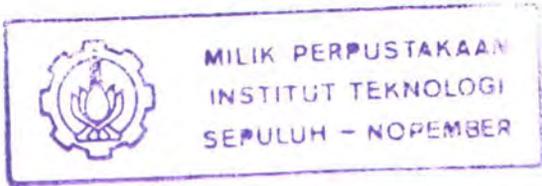
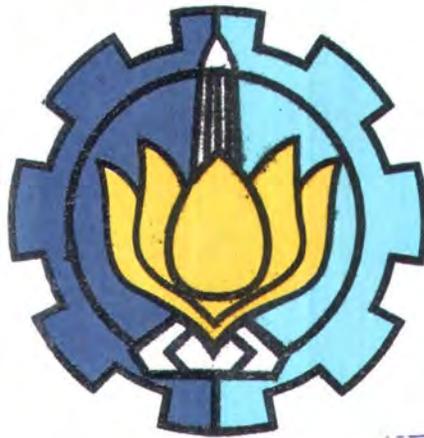


18.857/ITS/H/2003



PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK PENYARING HALAMAN INTERNET BERDASARKAN TEKS

TUGAS AKHIR



RSIF
005.1
Wib
P-1
2003

PERPUSTAKAAN ITS	
Tgl. Terima	19-8-2003
Terima Dari	H/
No. Agenda Prp.	719088

Disusun Oleh :

NUR CAHYO WIBOWO
5198 100 069

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2003**

**PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK
PENYARING HALAMAN INTERNET BERDASARKAN TEKS**

TUGAS AKHIR

Diajukan Untuk Memenuhi Sebagian Persyaratan

Memperoleh Gelar Sarjana Komputer

Pada

Jurusan Teknik Informatika

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya

Mengetahui / Menyetujui

Dosen Pembimbing



Febriliyan Samopa, M.Kom S.Kom

NIP. 132 206 858

**SURABAYA
JULI 2003**

ABSTRAK

Internet adalah sebuah medium terpenting yang ditemukan manusia pada abad 20. Saat ini semakin banyak hal dalam kehidupan manusia modern yang diperantarai oleh medium internet. Internet telah pula melahirkan dua bentuk fenomena yang berlawanan. Di satu sisi, ia telah memudahkan sekian banyak pekerjaan dalam kehidupan masyarakat modern, dan memungkinkan orang-orang untuk menciptakan pelbagai aktivitas baru dengan metode yang jauh lebih efisien. Tapi di sisi lain, ia juga telah melahirkan sejumlah masalah sosial baru. Salah satu fakta adalah hasil survei Forester Research pada awal tahun 90-an, menunjukkan bahwa hampir 80% lalu lintas internet selalu mengarah ke situs-situs dewasa (porno). Kondisi tersebut setidaknya menjadi indikator terjadinya dekadensinya moral pengguna internet. Selain itu juga menyebabkan pemborosan pemakaian *resource* yang berhubungan dengan internet, umumnya berkenaan dengan bandwidth koneksi dan resource penyimpanan yang ada.

Aplikasi Proxy Guard dalam Tugas Akhir ini merupakan salah satu sarana untuk melakukan penyaringan terhadap halaman-halaman internet berdasarkan teks yang berbau pornografi. Metode penyaringan yang digunakan adalah metode penentuan konteks halaman. Dimana konteks sebuah halaman internet tidak hanya ditentukan oleh sebuah kata kunci yang ditemukan, akan tetapi perlu dilakukan pencarian terhadap kata kunci penguat atau pelemahnya. Lebih jauh lagi juga perlu ditentukan bobotnya.

Hasil uji coba membuktikan bahwa aplikasi Proxy Guard ini mampu menjalankan fungsinya sebagai sebuah *proxy server* sekaligus sebagai penyaring halaman internet yang cukup efektif. Selain itu aplikasi ini juga memberikan sarana-sarana kemudahan untuk administrasi sebuah *proxy server*.

KATA PENGANTAR

Alhamdulillah Rabbal 'Alamin. Segala puji syukur ke hadirat Allah Subhanahu Wa Ta'ala, karena hanya dengan kehendak dan kuasa-Nya, penulis dapat menyelesaikan pembuatan tugas akhir yang berjudul **“PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK PENYARING HALAMAN INTERNET BERDASARKAN TEKS”**.

Tugas akhir dengan beban 4 SKS ini disusun dan diajukan sebagai salah satu syarat untuk menyelesaikan program Strata Satu (S1) pada jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Di dalam Tugas Akhir ini mungkin masih terdapat kekurangan – kekurangan yang belum bisa penulis sempurnakan. Untuk itu saran dan masukan sangatlah penulis harapkan untuk perbaikan ke depan.

Surabaya, Juli 2003

Nur Cahyo W

Ucapan Terima Kasih

Penulis menyadari bahwasanya dalam menyelesaikan Tugas Akhir ini telah mendapat bantuan dan dukungan yang tidak sedikit dari berbagai pihak. Untuk itu penulis secara khusus mengucapkan terima kasih dan penghargaan yang sebesar-besarnya kepada:

1. Bapak dan Ibu tersayang di rumah yang senantiasa mengingatkan dan mendoakan supaya Tugas Akhir ini segera penulis selesaikan.
2. Mbak Tatik dan Mas Hari, Mbak Dewi dan Mas Ari serta Dear Phit yang tidak bosan-bosannya menanyakan kabar Tugas Akhir penulis.
3. Bapak Yudhi Purwananto M.Kom selaku Ketua Jurusan Teknik Informatika – ITS bersama Ibu Nanik S M.Kom selaku dosen wali yang giat memotivasi mahasiswanya yang mulai kelebihan jatah normal kuliah.
4. Bapak Febriliyan Samopa M.Kom selaku dosen pembimbing yang dengan sabar memberikan arahnya selama pembuatan Tugas Akhir ini dan bantuannya di waktu sidang ujian.
5. Segenap dosen, karyawan dan admins di jurusan Teknik Informatika – ITS.
6. Reza, Aby dan Rizki atas bantuan teknisnya. You give me so much.
7. Rekan-rekan COE seperjuangan yang jadi maju maupun yang harus menundanya. You are my inspiration.
8. Para pecinta Keadilan Sejahtera atas nasehat dan kebersamaannya. Kita sukseskan Tahun Pemenangan. Allohu Akbar !!!

DAFTAR ISI

ABSTRAK	iii
KATA PENGANTAR	iv
DAFTAR GAMBAR	viii
DAFTAR TABEL	ix
BAB I	1
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	5
1.3. Batasan Permasalahan	6
1.4. Tujuan dan Manfaat	6
1.5. Metodologi Pembuatan Tugas Akhir	7
1.6. Sistematika Pembahasan	8
BAB II	10
DASAR TEORI	10
2.1. Penyaringan Internet	10
2.1.1. Metode Penyaringan	10
2.1.2. Penyaringan Berdasarkan Isi Halaman	12
2.1.3. Penyaringan Berdasarkan Sumber	14
2.1.4. Penyaringan Isi Halaman di Server	14
2.2. HTML	17
2.3. Protokol HTTP	17
2.3.1. Metode <i>Request</i> HTTP	20
2.3.2. <i>Header</i> HTTP	21
2.3.3. Status HTTP <i>Response</i>	22
2.4. <i>Socket</i>	22
<i>Socket</i> dalam C#	23
2.5. <i>Thread</i>	26
<i>Multithreading</i>	27
2.6. Microsoft Visual C# .NET	28
2.7. Regular Expression	30
2.7.1. Regular expressions sebagai Sebuah Bahasa	31
2.7.2. Regular expressions dalam .NET Framework	32
2.7.3. Class-class Regular expression	33
BAB III	40
DESAIN SISTEM	40
3.1. Definisi Sistem	40
3.2. Perancangan Sistem	40
3.2.1. Use Case Diagram	41
3.2.2. Activity Diagram	42
3.2.3. Sequence Diagram	43

3.3. Implementasi Sistem	46
3.3.1. Desain Antarmuka	46
3.3.2. Proses Penyaringan.....	56
BAB IV.....	63
IMPLEMENTASI DAN UJI COBA.....	63
4.1. Implementasi	63
4.2. Skenario Uji Coba.....	63
4.2.1. Kemampuan Penyaringan.....	64
4.2.2. Perbandingan Waktu	67
4.3. Evaluasi	68
BAB V.....	70
KESIMPULAN DAN SARAN	70
5.1. Kesimpulan.....	70
5.2. Saran.....	71
DAFTAR PUSTAKA.....	73

DAFTAR GAMBAR

Gambar 2.1 Proses koneksi protocol HTTP	18
Gambar 2.2 Bagan koneksi socket	23
Gambar 3.1 Use case diagram aplikasi	41
Gambar 3.2 Activity diagram pemakai	42
Gambar 3.3 Sequence diagram aplikasi	43
Gambar 3.4 Sequence diagram penyaringan.....	45
Gambar 3.5 Tampilan tab Proxy Configuration.....	46
Gambar 3.6 Tampilan tab Filter Settings - Advertisements.....	48
Gambar 3.7 Tampilan tab Filter Settings – Porn Words	49
Gambar 3.8 Tampilan tab Filter Settings – Porn Sites.....	50
Gambar 3.9 Tampilan tab Filter Settings - Response.....	51
Gambar 3.10 Tampilan tab Request Monitor.....	52
Gambar 3.11 Tampilan tab Process Monitor.....	53
Gambar 3.12 Tampilan output hasil penyaringan request.....	54
Gambar 3.13 Tampilan output hasil penyaringan response	54
Gambar 3.14 Tampilan window Help	55
Gambar 3.15 Tampilan window About.....	55

DAFTAR TABEL

Tabel 1.1 Statistik hit www.unmer.ac.id	2
Tabel 1.2 Top 15 website dengan rating tertinggi	4
Tabel 1.3 Top 15 website dengan pemilih terbanyak	4
Tabel 2.1 Daftar metode request HTTP	21
Tabel 2.2 Daftar kode status response HTTP	22
Tabel 4.1 Hasil uji coba untuk situs porno	64
Tabel 4.2 Hasil uji coba untuk situs bukan porno	65
Tabel 4.3 Hasil uji coba situs netral	66
Tabel 4.4 Hasil uji coba angka batas rasio	67
Tabel 4.5 Hasil uji coba situs dominan teks	71
Tabel 4.6 Hasil uji coba situs dominan bukan teks	71



BAB I
PENDAHULUAN

BAB I PENDAHULUAN

Pada bab ini akan dijelaskan mengenai beberapa hal dasar dalam Tugas Akhir ini yang meliputi latar belakang, perumusan masalah, tujuan dan manfaat pembuatan Tugas Akhir serta metodologi dan sistematika pembuatan buku Tugas Akhir ini. Dari uraian dibawah diharapkan gambaran umum Tugas Akhir bisa dipahami dengan baik.

1.1. Latar Belakang

Dalam sejarah perkembangan teknologi media, internet adalah sebuah medium terpenting yang ditemukan manusia pada abad 20. Meskipun tingkat penyebarannya di seluruh dunia masih belum sebesar teknologi media lain seperti televisi atau telepon dan media cetak, tapi perkembangan teknologi internet sendiri telah jauh melampaui apa yang sebelumnya dicapai oleh media tradisional tersebut. Saat ini semakin banyak hal dalam kehidupan manusia modern yang diperantarai oleh medium internet. Dalam kalimat lain, internet bukan hanya sebuah medium komunikasi dan penyebaran informasi dalam pengertian konvensional yang selama ini kita kenal, melainkan juga sebuah dunia baru dengan bermacam-macam aktivitas kehidupan manusia di dalamnya.

Pada level kehidupan sosial, seperti setiap invensi teknologi yang lain, internet telah pula melahirkan dua bentuk fenomena yang berlawanan. Pada sisi yang satu, ia telah memudahkan sekian banyak pekerjaan dalam kehidupan

masyarakat modern, dan memungkinkan orang-orang untuk menciptakan pelbagai aktivitas baru dengan metode yang jauh lebih efisien. Tapi di sisi lain, ia juga telah melahirkan sejumlah masalah sosial baru.

Menurut survei yang pernah dilakukan oleh Forester Research pada awal tahun 90-an, hampir 80% lalu lintas internet selalu mengarah ke situs-situs dewasa (porno). Dari survei tersebut menunjukkan masyarakat internet begitu "menikmati" hadirnya situs-situs dewasa tersebut. Situs porno Private (www.private.com) kini mendapat tiga juta pengunjung setiap bulan. Situs-situs porno di negara tertentu tercatat sebagai 20 situs terpopuler. Di Inggris sendiri berada pada posisi ke-11 terpopuler, menurut British Internet Research Group MMXI.

Statistik hits www.unmer.ac.id melaporkan bahwa ternyata situs berbau pornografi juga menempati rating tertinggi di lingkungan masyarakat kampus.

Tabel 1.1 Statistik hit www.unmer.ac.id

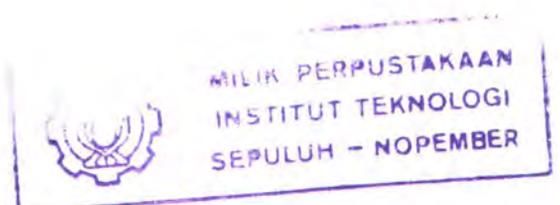
Hits		Search String
56	2.96%	sex bebas
26	1.37%	pergaulan bebas
3	0.16%	gambar seks kampus
3	0.16%	sex kampus
2	0.11%	gambar sex mahasiswa ukm
2	0.11%	gambar sex remaja
2	0.11%	mahasiswa sex
2	0.11%	seks bebas di kalangan mahasiswa
2	0.11%	seksualitas di kalangan mahasiswa

Laporan WebCounter menyebutkan bahwa situs di urutan rating pertama adalah yang berkategori '*amateur hardcore*' (penuh dengan tayangan seksual, hubungan seks, dengan berbagai gaya yang benar-benar liar dan tak pernah terbayangkan).

Mark Tiarra, pemilik TiarraCorp yang membuat daftar dan memberikan servis situs dewasa, memperkirakan bahwa sekitar 60 persen dari kesibukan lalu lintas Internet adalah karena akses paket-paket yang erat kaitannya dengan wanita telanjang bulat tersebut. Marc Kraft dari Adult Chamber of Commerce yang saat ini punya anggota sebanyak 100, bahkan berani menyebut angka 70 persen untuk yang dipastikan bersibuk dengan usahanya agar bisa menikmati sajian yang terlarang tersebut.

Marc Bell, yang mengelola kelompok Bell Technologies Group Ltd., dengan nilai investasi 17-juta USD, yang merupakan *host* terbesar bagi situs dewasa, menyatakan, bahwa banyak dan besarnya situs dewasa benar-benar bukan main. Bahkan untuk sebuah situs yang kelihatannya kecil-kecil saja telah mampu membikin sibuk lalu-lintas Internet, bahkan dibandingkan dengan situs non-telanjang yang besar sekalipun.

James Mann dari JMR Creations di Boston yang menjalankan situs Naughty Linx, yang menyediakan layanan link situs-situs seronok dengan tampilan seperti Yahoo, memperkirakan bahwa separuh dari situs komersial di Web adalah untuk sajian wanita telanjang dan seksualitas. Tahun lalu bahkan ia berani memperkirakan antara 80 sampai 90 persen.



Berikut ini adalah daftar situs dengan rating paling tinggi dan paling banyak dikunjungi dari indo-web. Dari data tersebut kembali dapat disimpulkan bahwa situs berbau pornografi tetap menjadi situs favorit.

Tabel 1.2 Top 15 website dengan rating tertinggi

Rating	# Votes	Resource
9.88	963	situs orang dewasa
9.08	26	Komik Saru MailSingkawang.com
8.95	8693	VCD Pomo Indonesia dan Asia
8.93	15	Indo Teen
8.74	31	Linna Home Collection
8.57	28	Galery Foto Artis Indonesia
8.33	15	Indo Teen
8.12	16	Popular Online
7.89	37	Indonesian Album
7.89	18	Cerita Seks Indonesia by KumpulanCerita.com
7.80	8342	Asian Model, Pondok Putri by Sanggrahan.org

Tabel 1.3 Top 15 website dengan pemilih terbanyak

Rating	# Votes	Resource
8.95	8693	VCD Pomo Indonesia dan Asia
7.80	8342	Asian Model, Pondok Putri by Sanggrahan.org
9.88	963	situs orang dewasa
7.89	37	Indonesian Album
8.74	31	Linna Home Collection
8.57	28	Galery Foto Artis Indonesia

9.08	26	Komik Saru MailSingkawang.com
7.89	18	Cerita Seks Indonesia by KumpulanCerita.com
8.12	16	Popular Online
8.33	15	Indo Teen
8.93	15	Indo Teen

Fakta-fakta di atas menunjukkan bahwa akses ke situs berbau pornografi sudah sedemikian besarnya. Hal tersebut menunjukkan indikasi terjadinya dekadensi moral masyarakat pengguna internet. Selain itu juga menimbulkan pengaruh terhadap kesibukan lalu lintas *resource* yang berhubungan dengan internet, baik yang langsung maupun tidak. Yang pada akhirnya akan menurunkan kinerja dan kegunaan *resource* tersebut baik dalam hal waktu maupun materi fisiknya.

Selain materi-materi berbau pornografi, *resource* yang berhubungan dengan internet juga perlu dicegah dari pemrosesan informasi yang tidak diperlukan, dalam hal ini adalah iklan. Seringkali iklan yang tidak dibutuhkan tampil secara tiba-tiba dan, tidak jarang pula iklan tersebut justru menampilkan materi berbau pornografi.

1.2. Perumusan Masalah

Permasalahan yang diangkat dalam Tugas Akhir ini adalah :

1. Bagaimana mendesain dan mengimplementasikan sebuah aplikasi *proxy server* untuk melayani *request* dari *client*.
2. Bagaimana mendesain aplikasi untuk mencegah request terlarang dari *client*.

3. Bagaimana mendesain aplikasi penilaian untuk menentukan apakah sebuah dokumen internet berformat html termasuk kategori porno atau tidak.
4. Bagaimana mendesain aplikasi yang konfigurasinya bisa diatur dengan mudah oleh pemakai.

1.3. Batasan Permasalahan

Dari permasalahan-permasalahan di atas, maka batasan dalam Tugas Akhir ini adalah:

1. File yang akan diproses adalah file teks berformat html.
2. Aplikasi yang dibangun menghubungkan antara komputer *client* dengan *proxy server* induk yang sudah *terinstall* di jaringan lokal.
3. Aplikasi berjalan sebagai sebuah program biasa di komputer yang terhubung ke jaringan, bukan sebagai sebuah program *dedicated server*.
4. Aplikasi dikembangkan dengan menggunakan Microsoft Visual C# .NET.

1.4. Tujuan dan Manfaat

Tujuan pembuatan Tugas Akhir ini adalah merancang dan membuat perangkat lunak penyaring dokumen internet berdasarkan teks untuk mencegah pengaksesan materi yang berbau pornografi maupun yang tidak diperlukan oleh pengguna sehingga diharapkan mampu menghemat *resource* yang ada.

1.5. Metodologi Pembuatan Tugas Akhir

Pembuatan Tugas Akhir ini terbagi menjadi beberapa tahapan sebagai berikut :

1. Studi Literatur

Pada tahap ini dilakukan pengumpulan dokumen-dokumen referensi yang diperlukan untuk merancang dan mengimplementasikan aplikasi Proxy Guard berbasis teknologi Microsoft Visual C# .NET.

2. Perancangan dan pembuatan perangkat lunak

Tahap ini merupakan tahap yang paling banyak memerlukan waktu karena meliputi beberapa proses yaitu perancangan perangkat lunak, pembuatan *prototype* perangkat lunak dan implementasi keseluruhan perangkat lunak.

3. Uji coba dan evaluasi perangkat lunak

Prototype aplikasi yang telah selesai ini nantinya akan diuji coba dan dievaluasi untuk kelayakan yang sesuai dengan tujuan.

4. Penyusunan Buku Tugas Akhir

Pada tahap terakhir ini disusun buku sebagai dokumentasi dari pelaksanaan Tugas Akhir. Dokumentasi ini juga dibuat sehingga memudahkan orang lain yang ingin mengembangkan sistem informasi tersebut, yang merupakan tahap akhir dari pengerjaan tugas akhir ini.

1.6. Sistematika Pembahasan

Sistematika pembahasan mengenai perangkat lunak yang dibuat dalam Tugas Akhir ini disusun dalam beberapa bab, yang dijelaskan sebagai berikut :

BAB 1 PENDAHULUAN

Bab ini berisi pendahuluan yang memuat latar belakang pembuatan aplikasi, perumusan masalah, pembatasan masalah, tujuan dan manfaat yang bisa diperoleh dalam Tugas Akhir ini serta sistematika dan metodologi yang digunakan dalam pembuatan buku Tugas Akhir.

BAB 2 DASAR TEORI

Bab ini berisi mengenai konsep dan teori pembelajaran yang menjadi landasan pembuatan Tugas Akhir. Dasar teori disini meliputi pengertian tentang dasar-dasar teknik penyaringan, internet serta tools yang digunakan.

BAB 3 DESAIN SISTEM

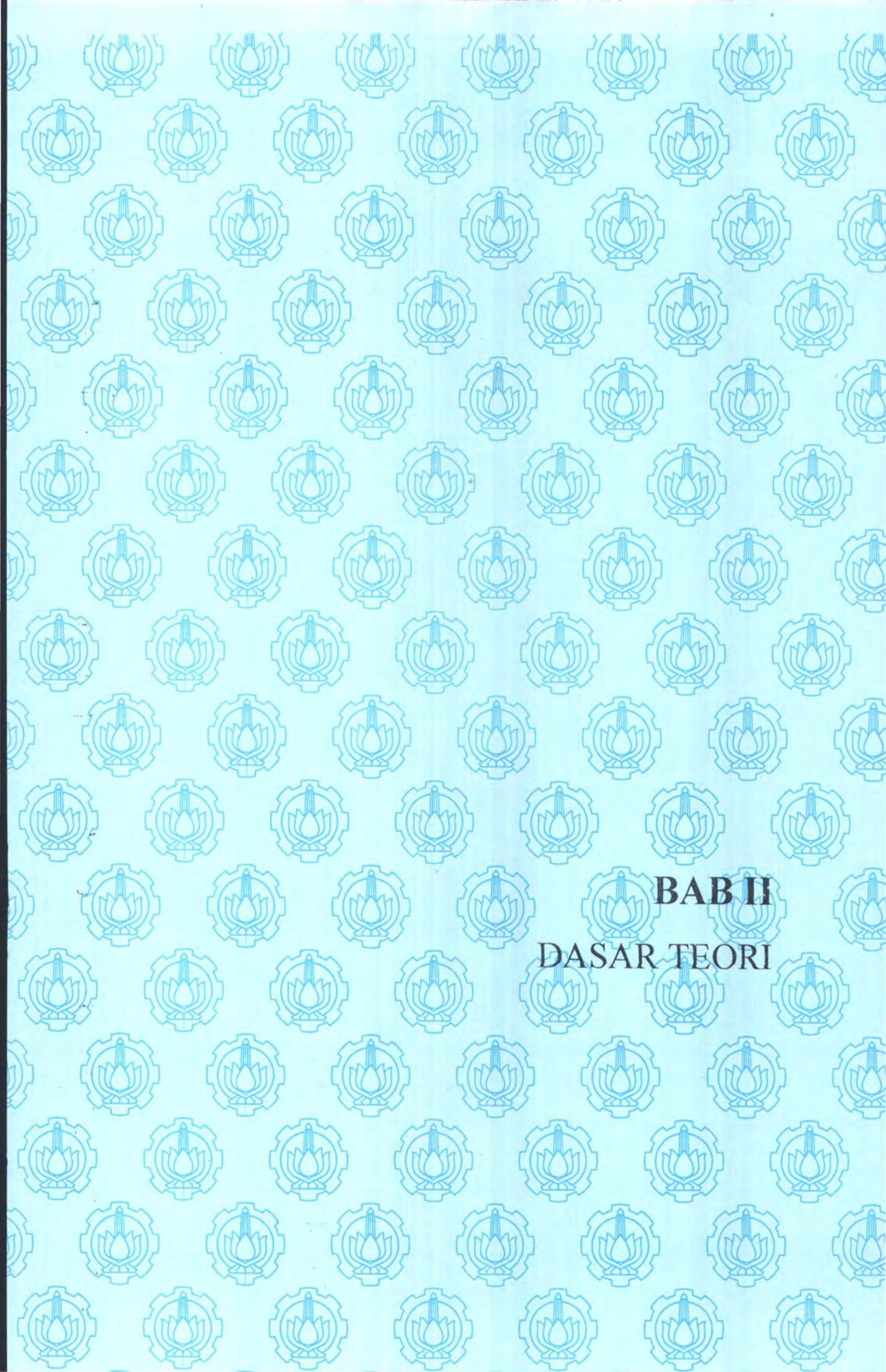
Bab ini berisikan mengenai perancangan sistem, yang meliputi deskripsi singkat aplikasi *penyaringan* ini serta deskripsi *class-class* dan fungsi-fungsi yang digunakan.

BAB 4 IMPLEMENTASI DAN UJI COBA SISTEM

Bab ini berisikan bagaimana implementasi aplikasi yang telah di buat berdasarkan desain sebelumnya. Juga dijelaskan mengenai hasil uji cobanya serta evaluasi terhadap aplikasi *penyaringan* tersebut.

BAB 5 KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan dan saran untuk pengembangan lebih lanjut dalam upaya memperbaiki kelemahan yang ada guna mendapatkan hasil yang lebih baik.



BAB II
DASAR TEORI

BAB II DASAR TEORI

2.1. Penyaringan Internet

Istilah penghalangan dan penyaringan seringkali dipakai sebagai sinonim untuk teknologi yang mencegah akses ke tipe atau bagian tertentu dari *resource* yang ada di internet. Di dalam tugas akhir ini istilah penghalangan berartikan suatu teknik yang dipakai untuk menghentikan lalu lintas internet berdasarkan alamat yang dituju. Sedangkan istilah penyaringan berartikan suatu teknik yang menghentikan akses ke sebuah alamat berdasarkan isi halaman alamatnya itu sendiri.

Sampai saat ini tidak ada satu teknik pun yang terbukti mampu memblokir 100% akses ke seluruh isi internet yang tidak diinginkan. Namun banyak diantaranya yang bisa cukup handal dalam kondisi pemakaian yang normal. Idealnya sebuah teknologi penghalangan dan penyaringan internet yang efektif haruslah mencatat semua kemungkinan jalan dimana isi internet bisa didistribusikan. Layanan penyaringan internet seharusnya mampu memblokir akses ke transfer file dan ruang *chatting* sebagaimana halnya kemampuannya menolak akses ke *website* terlarang yang sudah diketahui. [PG]

2.1.1. Metode Penyaringan

Ada tiga pendekatan dasar untuk menyaring isi internet:

Penyaringan Inklusi (daftar putih)

Metode ini hanya mengizinkan akses ke relatif sejumlah kecil situs yang sudah diketahui boleh untuk dikunjungi, dan memblok akses ke selainnya sampai ada pembuktian bahwa situs tersebut aman. Metode ini dilengkapi dengan daftar situs yang dinilai memenuhi kriteria yang diperbolehkan. Penyaringan ini bisa 100% efektif karena memang hanya situs aman saja yang bisa diakses. Kendala utama penyaringan ini adalah ukuran internet yang besar dan terus berkembang yang kesemuanya menuntut adanya penentuan boleh tidaknya untuk diakses.

Penyaringan Eksklusi (daftar hitam)

Penyaringan ini berdasarkan pada sebuah daftar hitam berisikan situs-situs yang telah diketahui terlarang untuk diakses. Teknik ini lebih umum dipakai dibandingkan penyaringan inklusi. Prinsip teknik ini adalah "*tidak bersalah sampai terbukti bersalah*" sehingga dengan demikian bisa dilakukan akses ke seluruh situs selama ia tidak termasuk ke dalam daftar hitam meskipun sama sekali belum dikenal. Metode ini mengandalkan kepada daftar hitamnya supaya tetap komperhensif dan selalu baru. Namun demikian tetap saja tidak dapat menjamin penggunaanya untuk masih bisa mengakses isi yang tidak diinginkan.

Penyaringan Isi Halaman

Metode terakhir ini mengizinkan akses ke seluruh internet namun akan dilakukan pengujian isi sebelum diteruskan ke pengguna. Teknik ini akan mencari kata-kata kunci di dalam halaman *web* atau mencari karakteristik lain yang diduga terindikasi sebagai isi halaman yang meragukan, misalnya gambar dengan ukuran

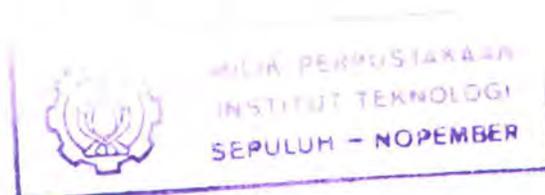
besar. Isi yang gagal memenuhi uji kebolehan tersebut akan diblok. Penyaringan isi cukup menjanjikan karena mampu melakukan klasifikasi secara dinamis setiap isi yang datang saat itu juga sehingga tidak diperlukan pengecekan *manual* sejumlah besar situs *web* dan pengguna pun tidak perlu memperbarui secara rutin daftar yang diperbolehkan dan yang tidak. Masalah yang dihadapi oleh pendekatan ini adalah sulitnya komputasi untuk menentukan boleh tidaknya akses diteruskan. [PG]

2.1.2. Penyaringan Berdasarkan Isi Halaman

Penyaringan ini menguji isi yang masuk dan *request* yang keluar untuk menentukan apakah termasuk akses yang diperbolehkan. Teknik ini mempunyai beberapa cara seperti pencarian kata-kata kunci, analisa gambar dan mencari karakteristik halaman *web* yang tidak diinginkan.

Penyaringan Kata Kunci

Penyaringan ini membaca isi halaman ketika sedang dikirim ke komputer pengguna sambil mencari kata-kata yang termasuk di dalam daftar hitam. Sebuah halaman akan diblok jika mengandung salah satu dari kata-kata yang terdapat di dalam daftar yang diblok. Teknik ini seringkali juga mengecek *request* sebelum dikirim keluar untuk mencegah pengguna supaya tidak memanfaatkan *search engine* untuk menemukan situs-situs yang mungkin mengandung isi yang tidak diinginkan tetapi belum termasuk di dalam daftar hitam yang sudah ada.



Penyaringan Frase

Penyaringan ini merupakan perluasan yang lebih rumit dari *penyaringan* kata kunci. Teknik ini tidak memperhatikan kata-kata yang berdiri sendiri, tetapi sebagai bagian sebuah frase. Ini memungkinkan klasifikasi yang lebih baik karena mampu membedakan frase *huge breasts*, dan *breast cancer* dalam konteks yang tepat. Namun cara ini masih mempunyai kendala seperti menentukan berapa banyak frase yang terlarang dibutuhkan sebelum halaman tersebut diblok, dan juga akan menjadi tak berguna untuk situs yang berbeda bahasanya. Dan kesulitan tambahannya adalah cara mendapatkan seluruh frase yang berbeda yang dianggap terlarang.

Penyaringan Analisa Gambar

Metode ini menguji gambar-gambar ketika dikirimkan ke pengguna. Pendekatan ini misalnya berusaha menemukan gambar yang berisi tubuh yang telanjang. Ini merupakan proses yang sangat sulit dan membutuhkan komputasi intensif sehingga bisa membedakan antara karya lukisan dengan gambar terlarang atau antara pornografi dengan foto keluarga yang sedang di pantai. Waktu yang dipakai untuk menguji gambar juga menjadi masalah tersendiri. Sebagai solusi maka umumnya gambar hasil analisisnya baru ditampilkan setelah menampilkan isi halaman ke pengguna. Halaman yang mengandung gambar yang tidak diinginkan akan ditambahkan ke dalam daftar hitam untuk jangka waktu ke depan.

2.1.3. Penyaringan Berdasarkan Sumber

Teknik ini memblokir *request* yang keluar maupun *response* yang masuk dari situs yang telah diketahui terlarang. Pemblokiran berdasarkan alamat yang terdapat di dalam *request* atau *response*, menggunakan baik URL yang ada di dalam pesannya ataupun alamat IP di dalam paket datanya. [PG]

2.1.4. Penyaringan Isi Halaman di Server

Teknologi penyaringan berdasarkan *server* yang paling paling terkenal adalah dengan memakai *proxy server*. *Proxy server* berada diantara pengguna dengan internet dan dapat menguji semua *request* serta mengembalikan isi *response*-nya. Semua *client* harus melalui *proxy server* supaya bisa mengakses internet. *Client* perlu dikonfigurasi perangkat lunaknya supaya mengarah ke sebuah *proxy server* agar dapat mengakses halaman *web* maupun file ftp.

Proxy server merupakan komputer berfungsi umum yang berperan sebagai sebuah wakil dari *server* nyata yang ada di internet. Tugas utamanya adalah sebagai *gateway*, mengecek lalu lintas masuk dan keluar, serta untuk meningkatkan waktu akses internet dengan jalan menyimpan salinan informasi yang sering diakses. *Proxy server* juga dapat berperang sebagai penyaring. *Proxy server* mampu mendapatkan *request web* dan dengan cepat melihat URL di dalam daftar situs atau halaman baik yang diijinkan maupun tidak. Dalam prakteknya, *proxy server* lebih sering memakai teknik penyaringan berdasarkan daftar hitam yang berisi bagian URL. *Proxy server* harus menangani sejumlah besar *request* setiap detiknya dan tidak akan memiliki cukup waktu untuk melakukan penyaringan berdasar isi yang cukup lambat eksekusinya. [PG]

Sifat umum sebuah *proxy server* adalah:

- Transparan; terlepas dari adanya penyaringan yang dilakukan pada *proxy*, maka keberadaan sebuah *proxy server* tidak boleh mempengaruhi hasil akhir. Pemakai akan mendapat *response* yang sama baik melalui koneksi langsung maupun koneksi melalui *proxy server*. *Proxy* yang tidak transparan berarti telah melakukan penulisan ulang URL di dokumen HTML dan redireksi HTTP sehingga kembali ke *proxy server*.
- Dikendalikan oleh *client*; maksudnya adalah bahwa menggaunakan *proxy* atau tidak semuanya tergantung kepada *client*.
- *Server* di internet tidak memperlmasalahkannya; *server web* di internet umumnya tidak memberikan *response* yang berbeda terhadap *request* dari *client* baik yang langsung maupun yang melalui *proxy server*. Protokol HTTP memang menyediakan informasi tentang keberadaan *proxy server* antara, namun seringkali informasi ini diabaikan oleh *server web*. [ARI]

Ada beberapa macam tipe *proxy server*, diantaranya adalah sebagai berikut:

- *Proxy server Generic Firewall*

Tipe ini adalah yang paling banyak dipakai. *Proxy* menangani lalu lintas *web*, termasuk HTTP, FTP dan protokol Gopher sama baiknya dengan penanganan protokol aman memakai SSL, seperti HTTPS dan SNEWS. *Proxy server generic* menyediakan layanan akses kontrol, penyaringan, pencatatan, dan *caching* yang bervariasi. *Proxy firewall* menerima *request* dari dalam *firewall*,

meneruskannya keluar ke internet, dan kemudian mengirimkan hasilnya kembali ke *client*. *Caching* biasa dilakukan oleh *proxy* ini, sehingga beberapa *request* sama sekali tidak perlu diteruskan ke *server* asli namun sebagai gantinya akan dilayani dari *cache*. Semua lalu lintas ke dan dari internet melalui *proxy firewall*; hanya ada satu pintu masuk untuk akses ke internet.

- *Proxy server Departemental*

Proxy ini pada prinsipnya juga merupakan *proxy server* tipe *generic firewall*, namun batasan pemakainya dipersempit: sebuah departemen tunggal dari sebuah perusahaan maupun kantor yang besar. Perangkat lunak yang dipakai mungkin sama, yang membedakannya adalah pengaturan konfigurasinya. Sebagai contoh misalnya sebuah departemen mungkin mempunyai kontrol akses yang lebih terbatas dibandingkan dengan departemen lainnya dan kontrol akses pada *proxy firewall* dapat pula berbeda antara *proxy* departemen satu dengan yang lain. *Proxy* departemental merupakan contoh dari *proxy* bertingkat.

- *Proxy server Personal*

Proxy ini merupakan *proxy server* yang diminimalisasikan hanya untuk pemakai individual. Umumnya digunakan pada *host* yang sama dengan program *client*. Layanan yang diberikan oleh *proxy personal* adalah *caching* lokal, *update cache* yang aktif, *polling* untuk perubahan dan peringatannya, manajemen daftar favorit personal dan pencarian lokal. [ARI]

2.2. HTML

Hypertext Markup Language (HTML) merupakan suatu skrip di mana kita bisa menampilkan informasi dan daya kreasi kita lewat Internet. HTML sendiri adalah suatu dokumen teks biasa yang mudah untuk dimengerti dibandingkan bahasa pemrograman lainnya, dan karena bentuknya itu maka HTML dapat dibaca oleh *platform* yang berlainan. Selain itu HTML merupakan bahasa pemrograman yang fleksibel dimana kita bisa meletakkan skrip dari bahasa lain.

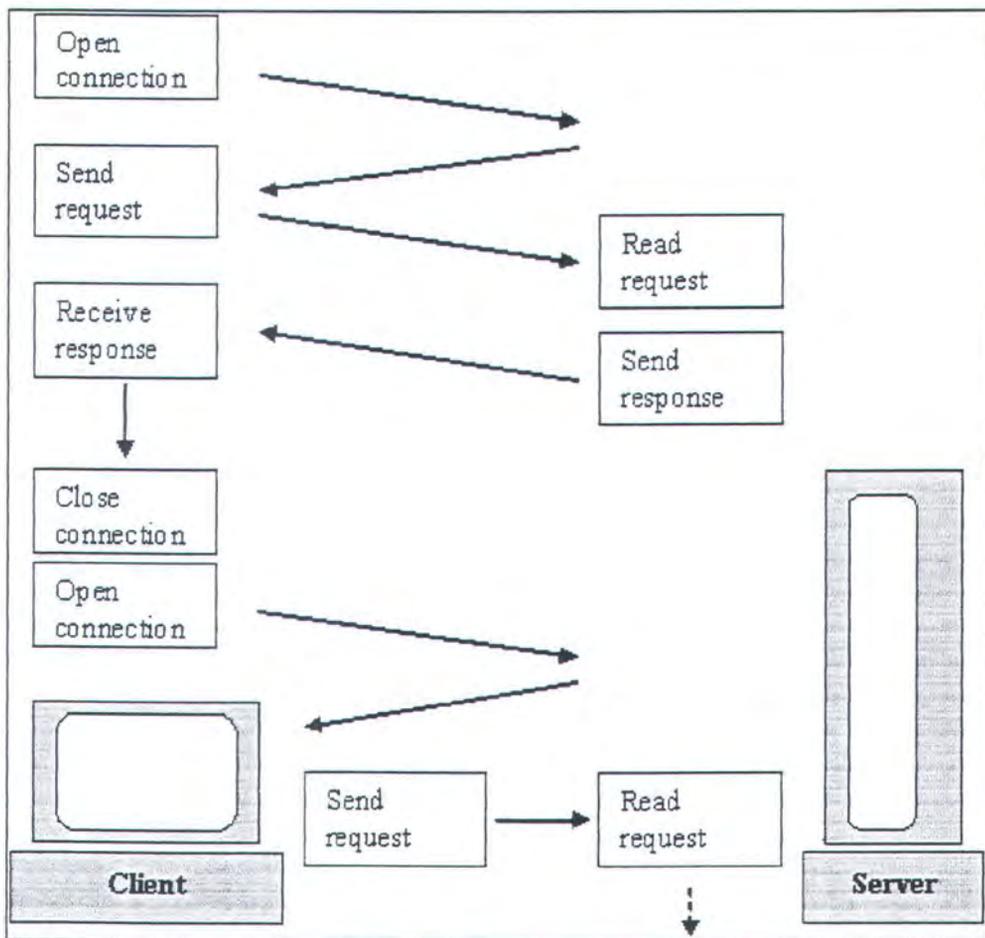
Hypertext dalam HTML berarti bahwa kita dapat menuju ke suatu tempat, misal website atau halaman homepage lain, dengan cara memilih *link* yang biasanya digarisbawahi atau diwakili oleh suatu gambar. Sedangkan markup language menunjukkan suatu fasilitas yang berupa tanda tertentu dalam skrip HTML dimana kita bisa mengatur judul, garis, table, gambar dan lain-lain dengan perintah khusus.

2.3. Protokol HTTP

Protokol HTTP (Hypertext Transfer Protocol) telah digunakan sejak tahun 1990. HTTP dimulai dengan versi 0.9 atau dengan lebih tepat lagi HTTP/0.9. Versi ini hanyalah protokol ringkas yang membolehkan pemindahan data melalui internet tanpa mengenali apakah jenis data yang dipindahkan. Setelah itu versi HTTP/1.0 dikeluarkan dan versi ini mampu memindahkan pesan dalam format "MIME", yang mengandung informasi mengenai data yang dipindahkan. Ini memungkinkan browser internet mengenali jenis data/ dokumen yang dipindahkan. Sebagai contoh, jika URL yang mengarahkan kepada file PDF diklik, maka browser internet kita secara otomatis akan membuka

plugin Acrobat Reader untuk menangani dokumen pdf tersebut.

Protokol HTTP yang terbaru adalah HTTP/1.1 (RFC 2616). Protokol HTTP berfungsi secara "*client-server*". Konsep "*client-server*" adalah, *client* akan meminta dokumen ("*resource*") dari *server*, *server* menerima dan memahami permintaan tersebut, dan membalas permintaan tersebut. *Client* kemudian akan memprosesnya.



Gambar 2.1 Proses koneksi protocol HTTP

Contoh:

Client request melalui browser dengan URL <http://it-kelantan.zzn.com>.

Maka selanjutnya *server web* Google menerima permintaan itu dan mencari dokumen yang diminta oleh *client* dan seterusnya membalas permintaan tersebut bersama dengan dokumen yang diminta. *Client* menerima balasan dari *server web* dan mulai menampilkan halaman tersebut

Untuk lebih detailnya secara teknis, lihat *request* dari *client* melalui protokol HTTP sebagai berikut

Request dari *client* :

```
GET / HTTP/1.1
User-Agent: Vbdownloader
Host: IT-Kelantan.zzn.com:80
Accept: */*
```

Balasan daripada *server web*

```
HTTP/1.1 302 Object moved
Server: Microsoft-IIS/4.0
Date: Fri, 20 Apr 2001 06:53:02 GMT
Set-Cookie: BIGipServerp-vzzn=486670858.20480.0000; path=/
Location: http://IT-Kelantan.zzn.com/email/english/dispatcher.asp
Content-Length: 176
Content-Type: text/html
Set-Cookie: ASPSESSIONIDGQGQHMJ=EGKMIHHDCHBBMFPJGJKAFJGK; path=/
Cache-control: private
```

Disini *server web* memberitahu *client* bahawa URL yang kita cari itu telah berpindah kepada lokasi baru yaitu <http://IT-Kelantan.zzn.com/email/english/dispatcher.asp>. Apabila *client* internet kita menerima balasan ini, ia akan meminta URL baru berdasarkan url yang di balas oleh *server*.

Request dari *client*:

```

GET      http://IT-Kelantan.zzn.com/email/english/dispatcher.asp
HTTP/1.1
User-Agent: Vbdownloader
Host: IT-Kelantan.zzn.com:80
Accept: */*

```

Balasan daripada *server*

```

HTTP/1.1 302 Object moved
Server: Microsoft-IIS/4.0
Date: Fri, 20 Apr 2001 06:57:40 GMT
Location: http://it-kelantan.zzn.com/email/login/login.asp
Content-Length: 169
Content-Type: text/html
Set-Cookie: ASPSESSIONIDGQGGQHMJ=MDMMIHHDGJOGGLKNIHBFKADOP; path=/
Cache-control: private

```

Sekali lagi *server* memberitahu bahawa lokasi URL yang diminta telah dipindahkan ke URL baru. Sekali lagi *client* akan melakukan permintaan baru berdasarkan URL yang dibalas oleh *server*. Dan barulah *server web* akan membalas dan menghantar URL yang diminta oleh *client*.

Banyak status yang diterima dari balasan *server web* tersebut. Antara lain yang terpenting adalah *Status-Code*, seperti 200 OK, 302 object moved dll, sesuai dengan RFC 2616

2.3.1. Metode *Request* HTTP

Kata pertama dalam HTTP *request* adalah *method*, yang mengindikasikan tindakan *request* tersebut. Tabel berisikan spesifikasi daftar *method* HTTP/1.1 yang biasa dipakai. [ARI]

Tabel 2.1 Daftar metode request HTTP

Metode	0.9	1.0	1.1	Deskripsi
GET	X	X	X	Mengambil sebuah <i>resource</i>
HEAD		X	X	Mengambil <i>metadata</i>
POST		X	X	Pengiriman form
PUT			X	<i>Upload</i> file
DELETE			X	Menghapus sebuah <i>resource</i>
TRACE			X	Melacak rantai <i>proxy</i>
OPTIONS			X	Pilihan <i>query</i> ke server

2.3.2. Header HTTP

Header HTTP dipakai untuk menyertakan informasi tambahan ke *request* maupun *response*. Spesifikasi HTTP/1.1 menyebutkan total sebanyak 46 *header*. *Header* tersebut dibagi menjadi empat kategori: *header* umum, *header request*, *header response* dan *header entitas*. *Header* umum terdapat baik pada *request* maupun *response*. *Header* entitas menjelaskan isi tubuh *request* maupun *response*. [ARI]

Request memiliki struktur umum sebagai berikut:

```
METHOD URL HTTP/version
... header umum ...
... header request ...
... header entitas (opsional) ...
baris kosong
... entitas request (jika ada) ...
```

Response mempunyai struktur yang mirip:

```
HTTP/version kode-status baris-alasan
```

... header umum ...
 ... header response ...
 ... header entitas (opsional) ...
baris kosong
 ... entitas resource (jika ada) ...

2.3.3. Status HTTP *Response*

Kode status HTTP dibagi menjadi lima kategori; range kode status berkisar antara 100-599. Digit pertama menunjukkan arti umum kode status; dua digit sisanya menjelaskan kondisi yang lebih detail. [ARI]

Tabel 2.2 Daftar kode status response HTTP

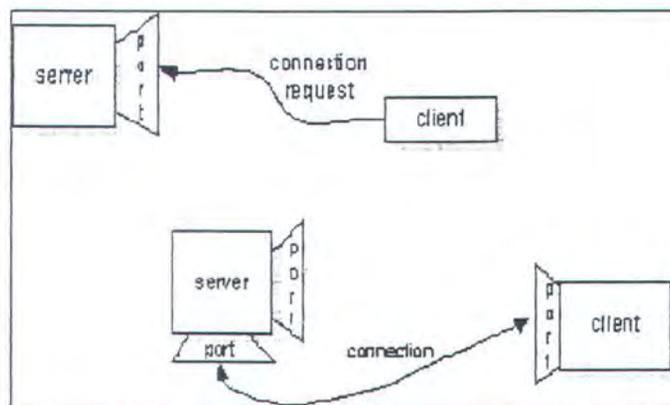
Kode	Kategori	Deskripsi
1xx	Informasi	Kode status aktual akan dikirimkan pada proses selanjutnya.
2xx	Sukses	Request berhasil diterima dan diproses.
3xx	Redirect	Kode status ini dipakai untuk meredirect client ke lokasi URL
4xx	Error di client	Menunjukkan terjadi error pada request client.
5xx	Error di server	Menunjukkan terjadi error pada sisi server.

2.4. *Socket*

Sebuah *socket* adalah satu ujung dari sebuah komunikasi dua arah antara dua program yang berjalan di jaringan. Sebuah *socket* diikat pada sebuah nomer *port*, sehingga lapisan TCP dapat mengidentifikasi aplikasi dimana data akan dikirimkan. *Socket* bisa dianalogikan seperti kotak-pos atau telepon, dimana *socket* adalah antarmuka antara user dan jaringan, seperti halnya kotak-pos, telepon masing-masing interface antara postal sistem, telepon sistem dan masyarakat.. *Socket* dapat dibentuk dan dihapus secara dinamik. Ketika *socket*

terbentuk maka salah satu parameternya menetapkan protokol yang digunakan, misalnya yang populer digunakan yaitu TCP/IP. [FE]

Sebelum *socket* dapat digunakan untuk *networking*, *socket* harus memiliki alamat yang dituju. Setelah *socket* dibentuk pada masing-masing komputer sumber dan tujuan, hubungan dapat dibangun antara keduanya. Setelah hubungan berhasil dibentuk, maka gabungan keduanya memiliki fungsi yang sama seperti pada pipa. [EDI]



Gambar 2.2 Bagan koneksi socket

Socket dalam C#

Class socket menyediakan kumpulan *method* dan *property* untuk komunikasi jaringan. *Class socket* memungkinkan untuk melakukan transfer data *asynchronous* maupun *synchronous* melalui protokol apapun yang terdapat dalam enumerasi *ProtocolType*.

Jika memakai protokol berorientasi koneksi, seperti TCP, maka *server* dapat menunggu koneksi menggunakan *method Listen*. *Method Accept*

memproses semua *request* koneksi yang masuk dan mengembalikan sebuah *socket* dimana bisa dipakai untuk mengkomunikasikan data dengan *remote host*. *Socket* ini bisa dipakai untuk memanggil *method Send* atau *Receive*. Sebelumnya *method Bind* harus dipanggil dulu sebelum memanggil *method Listen* jika menginginkan untuk menentukan alamat IP dan *port* lokal. *Property LocalEndPoint* mengidentifikasi alamat IP dan nomer *port* yang dipakai oleh *socket*. *Method Connect* dipanggil untuk menghubungkan ke *host* yang sedang menunggu. Untuk melakukan komunikasi data digunakan *method Send* atau *Receive*.

Setelah selesai mengirimkan dan menerima data, *socket* harus dimatikan dengan memanggil *method Shutdown*. Kemudian dilanjutkan dengan memanggil *method Close* untuk membebaskan semua *resource* yang sebelumnya dipakai oleh *socket*.

Contoh berikut ini menunjukkan bagaimana *class Socket* dapat dipakai untuk mengirimkan data ke sebuah *server* HTTP dan kemudian menerima *responsenya*.

```
using System;
using System.Text;
using System.IO;
using System.Net;
using System.Net.Sockets;
public class mySocket
{
    private static Socket connectSocket(string server, int port)
    {
        Socket s = null; IPEndPoint iphe = null;
        try
        {
            iphe = Dns.Resolve(server);
            foreach(IPAddress ipad in iphe.AddressList)
            {
                IPEndPoint ipe = new IPEndPoint(ipad, port);
            }
        }
    }
}
```



```

        Socket tmpS =
            new Socket(ipe.AddressFamily, SocketType.Stream,
ProtocolType.Tcp);
        tmpS.Connect(ipe);
        if(tmpS.Connected)
        {
            s = tmpS; break;
        }
        else
            continue;
    }
}
catch(SocketException e)
{
    Console.WriteLine("SocketException caught!!!");
}
catch(Exception e)
{ Console.WriteLine("Exception caught!!!"); }
return s;
}

private static string socketSendReceive(string server, int port)
{
    Encoding ASCII = Encoding.ASCII;
    string Get = "GET / HTTP/1.1\r\nHost: " + server +
        "\r\nConnection: Close\r\n\r\n";
    Byte[] ByteGet = ASCII.GetBytes(Get);
    Byte[] RecvBytes = new Byte[256];
    String strRetPage = null;
    Socket s = connectSocket(server, port);
    if (s == null)
        return ("Connection failed");
    s.Send(ByteGet, ByteGet.Length, 0);
    Int32 bytes = s.Receive(RecvBytes, RecvBytes.Length, 0);
    strRetPage = "Default HTML page on " + server + ":\r\n";
    strRetPage = strRetPage + ASCII.GetString(RecvBytes, 0,
bytes);
    while (bytes > 0)
    {
        bytes = s.Receive(RecvBytes, RecvBytes.Length, 0);
        strRetPage = strRetPage + ASCII.GetString(RecvBytes, 0,
bytes);
    }
    return strRetPage;
}

public static void Main(string[] args)
{
    string host; int port = 80;
    if (args.Length == 0)
        host = Dns.GetHostName();
    else
        host = args[0];
    string result = socketSendReceive(host, port);
    Console.WriteLine(result);
}
}

```

2.5. Thread

Sebuah *thread* adalah satuan dasar dimana sistem operasi mengalokasikan waktu prosesornya. Setiap *thread* menangani *exception handlers*, sebuah prioritas penjadwalan, dan satu set struktur dimana sistem akan menggunakannya untuk menyimpan konteks *thread* sampai ia terjadwal. Yang termasuk di dalam konteks *thread* adalah mesin, *register* dan *stack* yang berada dalam alamat prosesnya.

Sebuah sistem operasi *multitasking* membagi waktu *prosesor* untuk proses atau *thread* yang membutuhkannya. Sistem mengalokasikan potongan waktu *prosesor* ke setiap *thread* yang dieksekusi. *Thread* yang sedang dieksekusi akan ditangguhkan ketika potongan waktunya habis, kemudian *thread* lain ganti dijalankan. Status *thread* ketika terjadi pergiliran disimpan di dalam antrian. Lama potongan waktu *prosesor* bergantung pada sistem operasi yang bersangkutan dan juga prosesornya sendiri. Karena tiap potongan waktu itu begitu singkatnya maka *thread* ganda terlihat dieksekusi pada waktu yang bersamaan.

Pada umumnya *thread* digunakan untuk aplikasi yang:

- Melakukan operasi yang membutuhkan waktu cukup lama.
- Membedakan *task* yang mempunyai prioritas berbeda-beda.
- Mempertahankan antarmuka pemakai supaya tetap responsif, selama melakukan proses *background*.

Multithreading

Multithreading memungkinkan suatu aplikasi memproses lebih dari satu pekerjaan pada saat yang bersamaan [MSD]. Saat menggunakan *multithreading*, satu *thread* memproses antarmuka sementara *thread* lain melakukan kalkulasi-kalkulasi intensif atau memproses di latar belakang. Microsoft Visual Basic .NET memfasilitasi *multithreading*, sehingga kita dapat dengan mudah menggunakan kemudahan ini.

Karena sulitnya mendesain dan men-*debug* aplikasi *multithread*, Microsoft membuat konsep *Single-Threaded Apartment* (STA) di COM. Kode Visual Basic 6 selalu running di suatu STA, sehingga kode kita hanya akan mempunyai *single thread* untuk dipikirkan. Hal ini sepenuhnya menghindari keluaran-keluaran dengan data dan sumber yang dibagi, tetapi juga berarti kita tidak dapat mendapatkan keuntungan dari multi threading tanpa berulang kali menuju pengukuran-pengukuran ekstrem.

Pada .NET tidak ada yang seperti STA. Semua kode .NET berjalan dalam suatu *AppDomain*, yang memungkinkan terjadinya proses *multithreading*. Hal ini berarti kode Visual Basic .NET juga dapat berjalan di *AppDomain*, dan juga dapat mengambil keuntungan dari proses *Multithreading*. Jelas, setiap kali kita melakukan hal ini, kode harus ditulis dengan hati-hati untuk menghindari konflik antar *thread*.

Cara termudah untuk menghindari konflik antar *thread* adalah meyakinkan mereka untuk tidak sekalipun berinteraksi dengan data atau sumber yang sama. Walaupun tidak selalu mungkin, hal ini seharusnya menjadi tujuan desain dari

setiap aplikasi *multithread* untuk menghindari atau meminimasi penggunaan data atau sumber yang dibagi.

Tidak hanya hal ini menyederhanakan proses kode dan *debug*, tetapi juga menaikkan performa. Untuk menyelesaikan konflik antar *thread*, kita harus menggunakan teknik-teknik sinkronisasi yang sering menyebabkan *thread-thread* tersebut terblokir, atau berhenti sementara, sampai *thread* yang lain menyelesaikan prosesnya. Memblokir suatu *thread* berarti menyebabkannya *idle*, tidak melakukan apapun dan menurunkan performa.

Cara paling mudah untuk membuat proses background yang dapat berproses di *thread*-nya sendiri dengan datanya sendiri adalah untuk membuat suatu obyek khusus untuk proses background. Tujuan dilakukannya hal ini adalah baik, sepanjang dapat menyederhanakan pembuatan aplikasi *multithread*.

Jika *background thread* melakukan proses di dalam obyeknya sendiri, maka ia dapat memakai variable instan dari obyek tersebut (variable dideklarasikan dalam kelas) tanpa khawatir bahwa mereka akan dipakai oleh *thread* yang lain.

2.6. Microsoft Visual C# .NET

C# adalah sebuah bahasa pemrograman berorientasi obyek terbaru yang memungkinkan programmer membangun dengan cepat dan mudah aplikasi besar dalam *platform* Microsoft .NET. C# menyediakan *tools* dan layanan yang mendukung penuh komputasi dan komunikasi. C# didesain untuk pengembangan yang cepat bagi programmer C++ tanpa pengorbanan tenaga dan kendali yang

telah menjadi ciri dari C dan C++. *Frame work* yang ada memungkinkan komponen C# menjadi layanan web XML yang bisa diakses melalui internet, dari aplikasi apapun yang berjalan pada *platform* apapun juga.

Bahasa pemrograman yang dipakai meningkatkan produktivitas pengembang serta memberikan layanan untuk mengeliminasi kesalahan pemrograman yang dapat menambah biaya pembuatan. C# memungkinkan pembuatan web dengan cepat namun juga masih mampu mempertahankan kehandalan dan keluwesan yang dibutuhkan oleh para pengembang. [CSH]

Ekonomi web, dimana pesaing begitu banyak jumlahnya, memaksa bisnis untuk merespon tantangan kompetisi lebih cepat dari sebelumnya. Pengembang dituntut untuk memperpendek waktu siklus dan menghasilkan sebuah program yang direvisi secara menaik dari pada sebuah versi monumental tunggal. C# didesain dengan pertimbangan-pertimbangan tersebut. Bahasanya didesain untuk membantu pengembang supaya semakin sedikit baris kodenya dan semakin sedikit kemungkinan terjadi kesalahan.

Model baru untuk pembuatan aplikasi berarti akan lebih banyak lagi solusi yang membutuhkan kegunaan standar web yang sedang berkembang seperti Hypertext Markup Language (HTML), Extensible Markup Language (XML) dan Simple Object Access Protocol (SOAP). Sarana bantu pengembangan yang dibuat sebelum teknologi internet atau web jelas sudah ketinggalan jaman. Akibatnya tidak selalu bisa menyediakan pilihan terbaik untuk bekerja dengan teknologi web terbaru.

Seorang programmer C++ yang sudah ahli pun masih bisa melakukan kesalahan yang sederhana semisal untuk menginisialisasi variabel. Kesalahan kecil ini bisa menghasilkan masalah tidak terduga yang tetap tidak akan diketahui sampai jangka waktu yang lama. Sekali sebuah program sudah diimplementasikan, maka perbaikannya akan sangat menghabiskan energi meski hanya karena sebuah kesalahan yang sangat sederhana.

Desain modern C# mengeliminasi kesalahan umum pemrograman C++.

Misalnya:

- Koleksi sampah membebaskan programmer dari beban pengaturan memori manual.
- Variable dalam C# secara otomatis diinisialisasi oleh lingkungannya.
- Variable memiliki tipe data yang aman.

2.7. Regular Expression

Regular expression adalah sebuah pola teks yang terdiri dari karakter-karakter biasa (contohnya huruf a sampai z) dan karakter-karakter khusus (karakter meta). Pola tersebut menggambarkan satu atau lebih string yang cocok ketika pencarian teks.

Regular expression dapat digunakan untuk:

- Menguji sebuah pola yang terdapat di dalam sebuah *string*.
- Mengganti teks.
- Mengambil *substring* dari sebuah *string* berdasarkan kecocokan polanya.

Regular expression menyediakan metode yang kuat, fleksibel dan efisien untuk memproses teks [MSD]. Notasi pencocokan pola yang luas memungkinkan teks dengan ukuran besar dapat diparsing dengan cepat untuk menemukan pola karakter tertentu; untuk mengekstrak, mengedit, mengganti, atau menghapus *substring* teks; atau menambahkan *string* terekstraksi ke dalam koleksi untuk menghasilkan laporan. Untuk banyak aplikasi yang berhubungan dengan *string* (misalnya pemrosesan HTML, *parsing* log file, dan *parsing* HTTP header), *regular expression* merupakan *tool* yang diperlukan.

2.7.1. Regular expressions sebagai Sebuah Bahasa

Bahasa *regular expression* didesain dan dioptimasi untuk memanipulasi teks. Bahasanya terdiri dari dua tipe karakter dasar: karakter teks literal (normal) dan metakarakter-metakarakter. Kumpulan metakarakter itulah yang memberikan kekuatan memproses bagi *regular expression*.

Metakarakter `?` dan `*` adalah metakarakter yang sudah familiar dan digunakan dengan DOS file sistem untuk mewakili suatu karakter apapun atau kumpulan karakter. Perintah DOS `COPY *.DOC A:` memerintahkan sistem untuk menyalin file apa saja dengan ekstensi `.DOC` ke dalam disket di *drive* A. Metakarakter `*` mewakili nama file apa saja didepan ekstensi `.DOC`. *Regular expression* memperluas ide dasar ini dan menyediakan sekumpulan besar metakarakter yang memungkinkan untuk menjelaskan ekspresi pencocokan teks yang sangat kompleks dengan beberapa karakter relatif.

Sebagai contoh, *regular expression* `\s2000`, ketika diterapkan pada *body* teks, cocok dengan semua yang mengandung string “2000” yang didahului oleh karakter spasi-putih apa saja, seperti spasi atau *tab*.

Regular expression juga menyediakan pencarian yang lebih kompleks. Sebagai contoh, *regular expression* `(?<char>\w)\k<char>`, menggunakan grup bernama dan backreferencing (dijelaskan pada bagian berikutnya), mencari karakter berdekatan yang berpasangan. Ketika diterapkan pada *string* “I’ll have a double tall latte”, ditemukan kecocokan pada kata “I’ll”, “tall”, dan “latte”.

Bagian selanjutnya menjelaskan kumpulan metakarakter yang menjelaskan bahasa *regular expression* dalam *.NET Framework* dan menunjukkan bagaimana menggunakan *class regular expression* untuk mengimplementasikan *regular expression* dalam aplikasi yang akan dibuat.

2.7.2. Regular expressions dalam .NET Framework

Regular expressions Microsoft *.NET Framework* meliputi fitur *regular expression* paling populer seperti yang ada di Perl dan awk. Dengan desain yang disesuaikan dengan *regular expression* dari Perl 5, *Regular expression .NET Framework* meliputi fitur yang belum pernah ada di implementasi lain, yaitu pencocokan dari kanan ke kiri (*right-to-left matching*) dan kompilasi sambil jalan (*on-the-fly compilation*).

Kelas-kelas *Regular expressions* Microsoft *.NET Framework* adalah bagian dari *base class library* dan bisa digunakan dengan bahasa apa saja atau *tool*

yang menggunakan *Common Language Runtime* (CLR), termasuk ASP.NET dan Visual Studio.NET

2.7.3. Class-class Regular expression

Bagian ini menjelaskan kelas-kelas *regular expression* .NET Framework [MSD]:

Regex

Class Regex mewakili *regular expression* yang tidak dapat diubah (*read-only*) *Regex* juga berisi metode statis yang memungkinkan penggunaan *class-class regular expression* lain tanpa membuat (*instantiating*) obyek dari *class-class* lain secara eksplisit.

Match

Class Match mewakili hasil operasi pencocokan *regular expression*. Contoh berikut menggunakan metode *Match* dari kelas *Regex* untuk mengembalikan obyek dengan tipe *Match* dengan tujuan menemukan kecocokan pertama dalam *string* yang dimasukkan. Contoh ini menggunakan properti *Match.Success* dari kelas *Match* untuk mengindikasikan apakah telah ditemukan kecocokan.

```
Regex r = new Regex("abc");
Match m = r.Match("123abc456");
if (m.Success)
{
    Console.WriteLine("Found match at position " + m.Index);
}
```

MatchCollection

Class MatchCollection mewakili urutan pencocokan *non-overlapping* yang berhasil. Kumpulannya tidak dapat diubah (*read-only*) dan tidak mempunyai *public constructor*. *Instances* dari *MatchCollection* dikembalikan oleh properti *Regex.Matches*.

Contoh dibawah ini menggunakan metode *Matches* dari kelas *Regex* untuk mengisi sebuah *MatchCollection* dengan semua kecocokan yang ditemukan pada *string* yang dimasukkan (*input string*). Contoh berikut menyalin kumpulan ke sebuah *string array* yang menyimpan setiap kecocokan dan *integer array* yang menunjukkan posisi dari masing-masing kecocokan.

```
MatchCollection mc;
String[] results = new String[20];
int[] matchposition = new int[20];
Regex r = new Regex("abc");
mc = r.Matches("123abc4abcd");
for (int i = 0; i < mc.Count; i++)
{
    results[i] = mc[i].Value;
    matchposition[i] = mc[i].Index;
}
```

GroupCollection

Class GroupCollection mewakili koleksi grup yang didapat dan mengembalikan kumpulan grup tersebut dalam sebuah kecocokan tunggal (*single match*). Koleksi ini tidak dapat diubah (*read-only*) dan tidak mempunyai *public constructor*. *Instances* dari *GroupCollection* dikembalikan ke dalam koleksi yang dikembalikan oleh properti *Match.Groups*.

Contoh aplikasi *console* berikut ini menemukan dan mencetak sejumlah grup yang didapatkan oleh *regular expression*. Contoh mengenai bagaimana mengekstrak masing-masing gambar dalam masing-masing anggota koleksi grup dapat dilihat dalam contoh `CaptureCollection` pada bagian berikutnya.

```
using System;
using System.Text.RegularExpressions;
public class RegexTest
{
    public static void RunTest()
    {
        // Menentukan grup "abc", "ab", dan "b".
        Regex r = new Regex("(a(b))c");
        Match m = r.Match("abdabc");
        Console.WriteLine("Number of groups found = " +
m.Groups.Count);
    }
    public static void Main()
    {
        RunTest();
    }
}
```

Contoh ini menghasilkan output sebagai berikut.

```
Number of groups found = 3
```

CaptureCollection

Class CaptureCollection mewakili *captured substrings* dan mengembalikan kumpulan *captures* yang dilakukan oleh satu grup *capturing*. Sebuah grup *capturing* bisa meng-*capture* lebih dari satu *string* dalam sebuah kecocokan yang disebabkan oleh *quantifiers*. Property *captures*, sebuah obyek dari *class CaptureCollection*, disediakan sebagai anggota dari *class-class Match* dan *Group* untuk memungkinkan akses terhadap kumpulan *captured substrings*.

Sebagai contoh, jika digunakan *regular expression* $((a(b))c)^+$ (dimana quantifier “+” menunjukkan satu atau lebih kecocokan) untuk meng-*capture* kecocokan dari string “abcabcabc”, *CaptureCollection* untuk setiap grup kecocokan dari *substrings* akan berisi tiga anggota.

Contoh aplikasi *concole* berikut ini menggunakan *regular expression* $(Abc)^+$ untuk menemukan satu atau lebih kecocokan dalam string “XYZAbcAbcAbcXYZAbcAb”. Contoh ini mengilustrasikan penggunaan properti *Capture* untuk mengembalikan grup-grup multipel dari *captured substrings*.

```
using System;
using System.Text.RegularExpressions;
public class RegexTest
{
    public static void RunTest()
    {
        int counter;
        Match m;
        CaptureCollection cc;
        GroupCollection gc;
        Regex r = new Regex("(Abc)+");
        m = r.Match("XYZAbcAbcAbcXYZAbcAb");
        gc = m.Groups;

        Console.WriteLine("Captured groups = " +
gc.Count.ToString());
        for (int i=0; i < gc.Count; i++)
        {
            cc = gc[i].Captures;
            counter = cc.Count;
            Console.WriteLine("Captures count = " +
counter.ToString());
            for (int ii = 0; ii < counter; ii++)
            {
                Console.WriteLine(cc[ii] + " Starts at
character " +
                cc[ii].Index);
            }
        }
    }
}
```

```

    }

    public static void Main() {
        RunTest();
    }
}

```

Contoh ini menghasilkan output sebagai berikut.

```

Captured groups = 2
Captures count = 1
AbcAbcAbc  Starts at character 3
Captures count = 3
Abc  Starts at character 3
Abc  Starts at character 6
Abc  Starts at character 9

```

Group

Class Group mewakili hasil dari sebuah grup *capturing* tunggal. Karena *Group* dapat meng-*capture* nol, satu, atau lebih *string* dalam sebuah kecocokan (menggunakan *quantifiers*), *Group* berisi kumpulan obyek *Capture*. Karena *Group* merupakan turunan dari *Capture*, *substring* terakhir yang di-*capture* bisa diakses secara langsung (*instance Group* itu sendiri sama dengan item terakhir dari kumpulan yang dihasilkan oleh properti *Captures*). *Instances* dari *Group* dihasilkan oleh properti *Match.Groups(groupnum)*, atau *Match.Groups("groupname")* jika digunakan konstruksi pengelompokan “(?<namagroup>)”.

Contoh berikut ini menggunakan konstruksi pengelompokan bersarang (*nested grouping construct*) untuk meng-*capture substrings* ke dalam *groups*.

```
int[] matchposition = new int[20];
```

```
String[] results = new String[20];
Regex r = new Regex("(a(b))c");
Match m = r.Match("abdabc");
for (int i = 0; m.Groups[i].Value != ""; i++)
{
    results[i]=m.Groups[i].Value;
    matchposition[i] = m.Groups[i].Index;
}
```

Contoh ini menghasilkan output sebagai berikut.

```
results[0] = "abc"    matchposition[0] = 3
results[1] = "ab"    matchposition[1] = 3
results[2] = "b"     matchposition[2] = 4
```

Contoh berikut menggunakan konstruksi pengelompokan yang diberi nama untuk meng-*capture substrings* dari sebuah *string* yang berisi data dalam format "DATANAME:VALUE" yang dipisahkan oleh *regular expression* pada kolon (":").

```
Regex r = new Regex("^ (?<name>\\w+) : (?<value>\\w+) ");
Match m = r.Match("Section1:119900");
```

Regular expression ini menghasilkan output sebagai berikut.

```
m.Groups["name"].Value = "Section1"
m.Groups["value"].Value = "119900"
```

Capture

Kelas Capture berisi hasil dari *subexpression capture* tunggal. Contoh berikut ini melakukan pengulangan kumpulan *Group*, mengekstrak kumpulan *Capture* dari masing-masing anggota *Group*, dan menentukan variabel *posn* untuk

posisi karakter dalam *string* yang asli dimana masing-masing *string* telah ditemukan dan variabel *length* untuk panjang dari masing-masing *string*.

```
Dim r As Regex
Dim m As Match
Dim cc As CaptureCollection
Dim posn, length As Integer

r = New Regex("(abc)*")
m = r.Match("bcabcabc")
Dim i, j As Integer
i = 0
While m.Groups(i).Value <> ""
    cc = m.Groups(i).Captures ' Grab the Collection for
Group(i).
    For j = 0 To cc.Count - 1
        posn = cc(j).Index ' Position of Capture object.
        length = cc(j).Length ' Length of Capture object.
    Next j
    i += 1
End While
```



BAB III
DESAIN SISTEM

BAB III DESAIN SISTEM

3.1. Definisi Sistem

Aplikasi penyaringan Proxy Guard dikembangkan dengan teknik penyaringan berdasarkan isi halaman. Kemudian digabungkan dengan penyaring berdasarkan URL untuk mencegah *request* yang tidak diijinkan. Aplikasi ini menerapkan metode pencocokan kata kunci. Namun dilakukan perluasan untuk menentukan konteks isi dari sebuah dokumen halaman internet. Selain itu juga dilakukan pembobotan sebagai nilai batasnya. Sehingga dengan kombinasi tersebut diharapkan mampu membangun sebuah sistem penyaring yang handal.

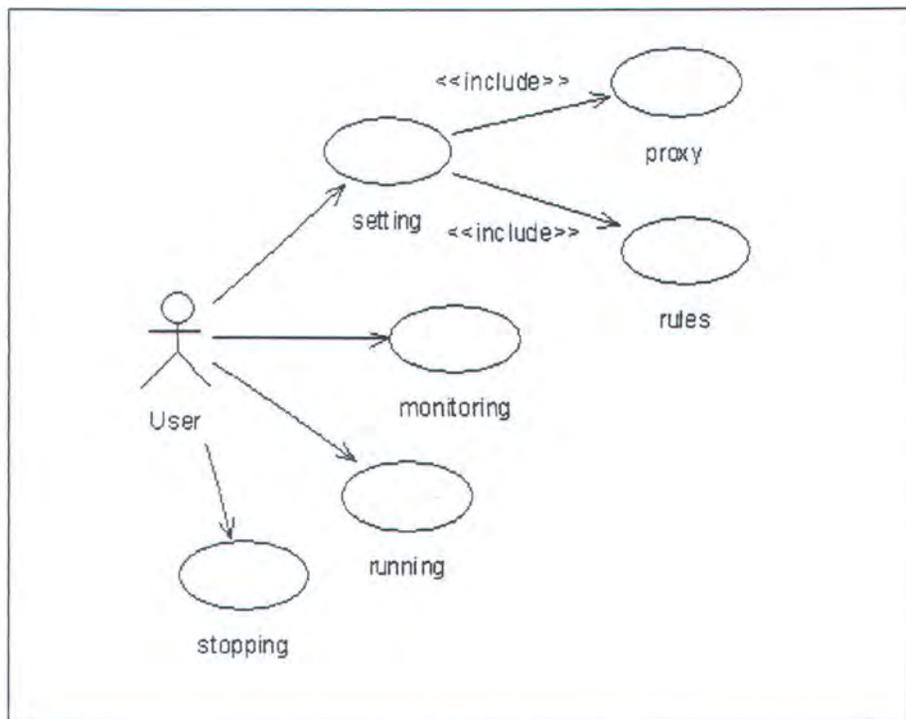
Sebagaimana layaknya sebuah *proxy server*, maka aplikasi Proxy Guard berjalan diantara *client* dan *proxy server* yang lebih tinggi struktur hirarki jaringannya. Aplikasi Proxy Guard akan menerima seluruh *request* yang dikirimkan oleh *client* untuk diteruskan ke *parent proxy server* di atasnya. Diantara proses penerimaan *request* sampai dilakukannya pengiriman *response* ke *client* dilakukanlah penyaringan, baik terhadap *request*-nya itu sendiri maupun terhadap *response* yang diterima dari *server* di internet..

3.2. Perancangan Sistem

Perancangan sistem berisikan penjelasan tentang use case diagram, activity diagram dan sequence diagram sistem. Sequence diagram terdiri dari dua bagian yaitu sequence untuk aplikasi dan sequence untuk penyaringan.

3.2.1. Use Case Diagram

Diagram berikut menjelaskan tentang aktivitas yang bisa dilakukan oleh pemakai aplikasi Proxy Guard.



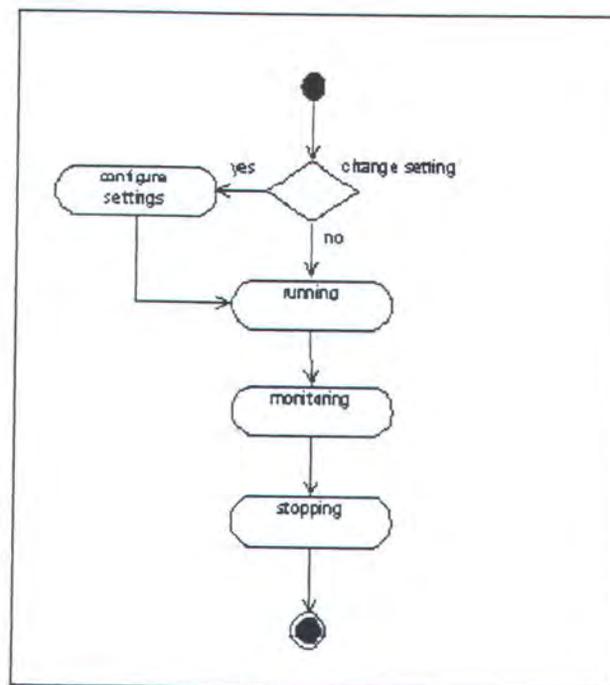
Gambar 3.1 Use case diagram aplikasi

Setting merupakan aktivitas pemakai di dalam mengatur konfigurasi aplikasi sebelum dijalankan. Setting yang bisa dilakukan pada konfigurasi *proxy* maupun konfigurasi *rules*. *Running* merupakan aktivitas pemakai untuk mengaktifkan aplikasi. Sedangkan *monitoring* merupakan aktivitas pemakai ketika mengamati proses penanganan *request* dari *client* yang sedang berlangsung. *Stopping* dilakukan untuk mengakhiri pemakaian aplikasi.

3.2.2. Activity Diagram

Diagram ini menggambarkan urutan proses yang bisa dilakukan oleh pemakai. Penggunaan aplikasi dimulai dengan sebuah pilihan untuk melakukan konfigurasi settings yang dibutuhkan. Nilai *defaultnya* sudah ada. Namun nilai-nilai tersebut masih bisa diganti oleh pemakai.

Jika pemakai tidak menghendaki perubahan konfigurasi *setting* maka pemakai bisa langsung menjalankan aplikasi. Setelah aplikasi dijalankan maka pemakai bisa memantau request dari *client* yang masuk, maupun memantau proses latar yang sedang berjalan. Setelah fungsi *proxy* tidak diperlukan lagi maka pemakai bisa mematikannya.



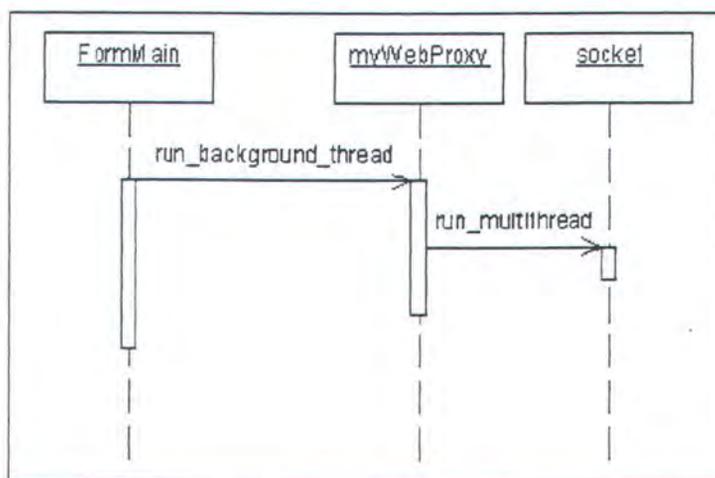
Gambar 3.2 Activity diagram pemakai

3.2.3. Sequence Diagram

Diagram ini menunjukkan urutan proses yang dikerjakan di dalam aplikasi Proxy Guard. Ada dua diagram utama yang perlu dijelaskan, yaitu *sequence diagram* untuk proses muka dan *sequence diagram* untuk proses latar, dalam hal ini proses penyaringannya.

3.2.3.1. Sequence Diagram Aplikasi

Sequence diagram proses muka dimulai dari objek FormMain yang merupakan *class* antarmuka. Di dalam *class* ini terdapat *event* untuk mengaktifkan *class* myWebProxy yang akan menjadi proses latar. *Class* myWebProxy dibangun dengan konsep *thread* sehingga ketika ia sedang berjalan tetap tidak akan mengganggu responsivitas antarmuka pemakai. *Class* myWebProxy berfungsi untuk mengaktifkan *socket* untuk menangani setiap *request* dari *client* yang masuk.



Gambar 3.3 Sequence diagram aplikasi

Request dari *client* bisa jadi datang secara bersama-sama dan dalam jumlah yang cukup banyak. Untuk itu penanganannya membutuhkan kembali *thread*, sehingga pemrosesan *request* bisa dilakukan secara simultan tanpa harus menunggu proses yang awal selesai dikerjakan terlebih dahulu oleh aplikasi.

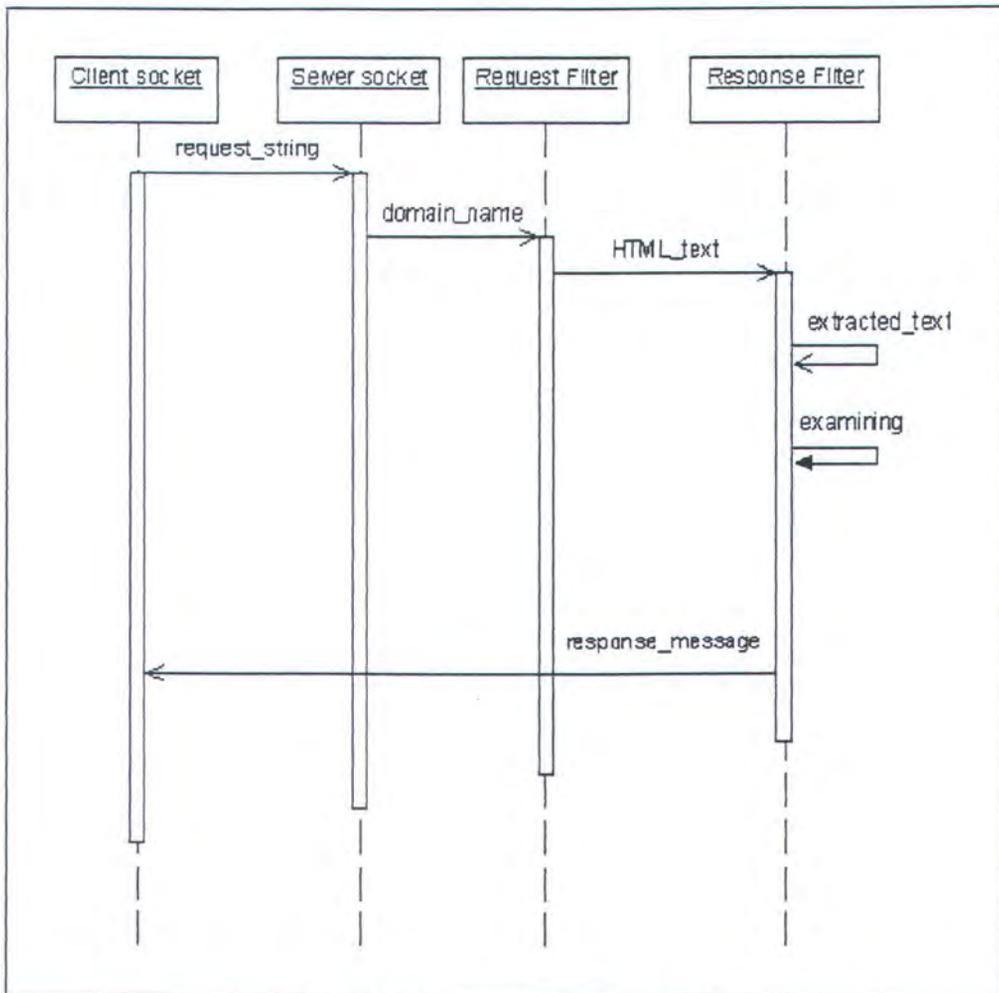
3.2.3.2. Sequence Diagram Penyaringan

Sequence diagram yang kedua adalah diagram proses penyaringan. Proses ini dimulai ketika *string request* dikirimkan oleh *client* ke aplikasi Proxy Guard. *String request* yang masuk kemudian diambil informasi nama *domainnya*.

Nama *domain* ini kemudian akan dijadikan masukan bagi subproses yang menangani penyaringan *request*. Nama *domain* akan dicocokkan dengan daftar hitam alamat situs yang sudah ada. Jika terjadi kecocokan maka proses penyaringan akan dihentikan. Dan *request* dari *client* akan *redirect* ke sebuah URL dokumen html yang ada di komputer lokal. Proses *request* akan selesai sampai di sini.

Namun jika nama *domain* tidak termasuk dalam daftar hitam, maka proses penanganan *request* akan terus dilanjutkan. Aplikasi akan meneruskan *request* dari *client* ke *server* di internet. Kemudian *response* berupa teks yang datang akan ditampung sementara oleh aplikasi, sedangkan *response* berupa non teks akan langsung diteruskan ke *client*.

Setelah semua teks diterima maka dilakukan proses ekstraksi untuk menghapus semua informasi maupun karakter-karakter yang tidak diperlukan sehingga pada akhirnya akan diperoleh teks yang merupakan isi sebuah file html.



Gambar 3.4 Sequence diagram penyaringan

Himpunan teks tersebut kemudian diproses untuk ditentukan konteksnya, termasuk kategori porno ataukah tidak. Proses dimulai dengan menghitung perbandingan antara frekuensi ditemukannya kata kunci dengan jumlah total kata yang terdapat di dalam sebuah halaman web.

Jika tidak terpenuhi maka halaman web dinilai cukup aman untuk diteruskan ke *client*. Namun jika ternyata terpenuhi kondisinya maka akan

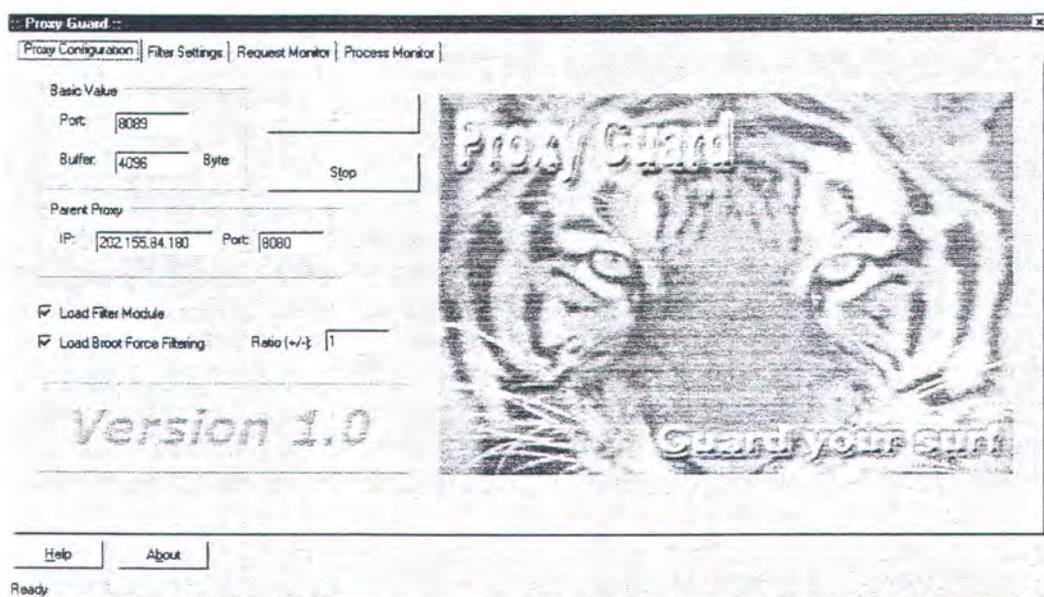
dilanjutkan dengan proses pencocokan kata kunci kedua yang berperan sebagai penentu konteks sebuah halaman.

Jika frekuensi kata kunci kedua tidak terpenuhi maka halaman web dinilai cukup aman untuk dikirimkan ke *client*. Namun jika ternyata memenuhi kriteria maka aplikasi akan melakukan penulisan ulang *response* asli yang diterima dari *server* di internet. Hasil penulisan ulang ini yang akan dikirimkan ke *client*. *Response* modifikasi ini berisikan informasi kenapa *request* tidak dilanjutkan untuk diproses. Sampai di sini proses penyaringan *response* berakhir.

3.3. Implementasi Sistem

3.3.1. Desain Antarmuka

Tab Proxy Configuration

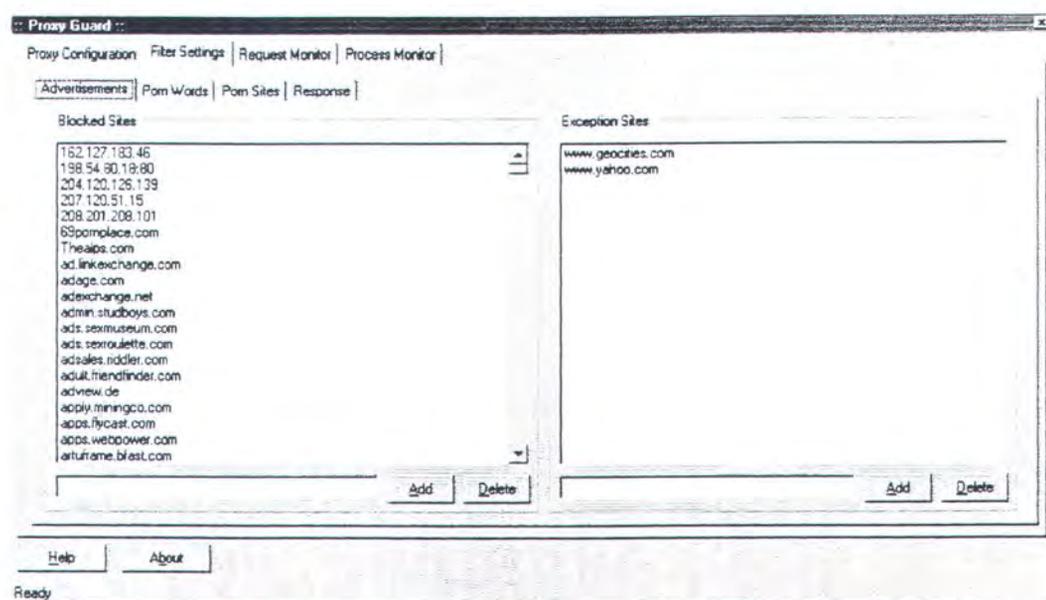


Gambar 3.5 Tampilan tab Proxy Configuration

Form utama aplikasi Proxy Guard dibangun dengan memakai komponen kontrol Tab. Terdiri dari empat buah tab, yaitu tab Proxy Configuration, tab Filter Settings, tab Request Monitor dan tab Process Monitor.

Tab Proxy Configuration berfungsi sebagai antarmuka untuk menerima input yang berhubungan dengan konfigurasi aplikasi. Di dalam tab ini bisa diatur nomor port yang akan dipakai, serta besarnya ukuran cache yang akan digunakan. Pengaktifan *proxy* bisa dilakukan dengan menekan tombol Start. Untuk menghentikannya dilakukan dengan menekan tombol Stop. *Proxy* induk juga bisa diganti-ganti alamat IP dan portnya sesuai dengan kondisi jaringan yang ada. Di bagian bawah bisa dilakukan pemilihan opsi pengaktifan *penyaringan* atau tidak. Perubahan pada pilihan *checklist* tersebut ketika *proxy* dalam kondisi *running* akan otomatis mengubah jalannya program latar secara waktu nyata. Jadi tidak perlu menghentikan *proxy* untuk melakukan *restart*.

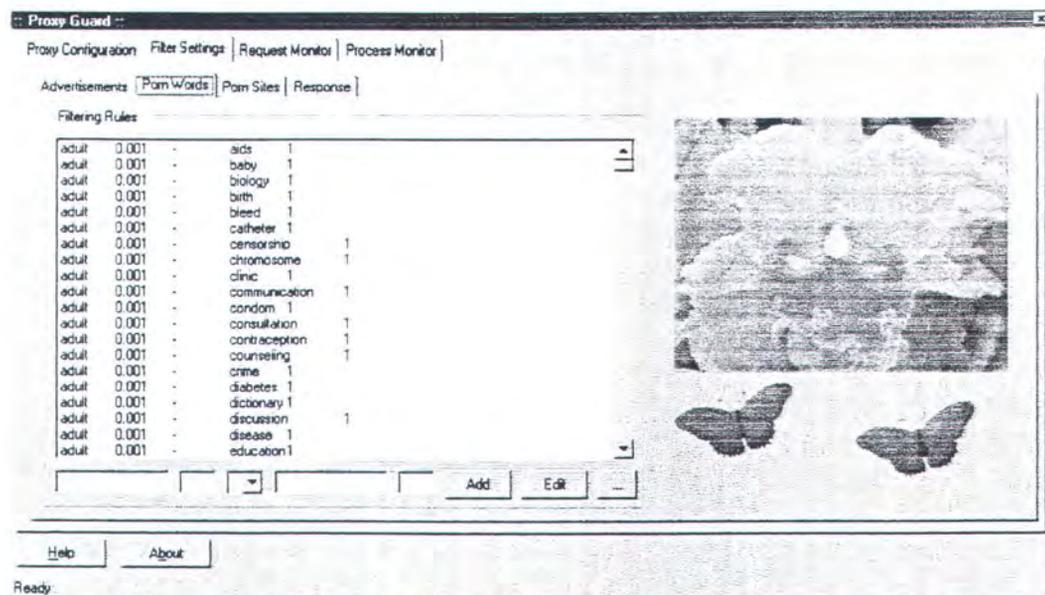
Tab Filter Settings – Advertisements



Gambar 3.6 Tampilan tab Filter Settings - Advertisements

Tab Advertisement berisikan dua buah kotak kelompok. Kelompok pertama berisikan manajemen daftar situs yang biasa memunculkan pop up iklan. Sedang yang satunya berisikan manajemen daftar situs perkecualian. Masing kelompok dilengkapi dengan fasilitas untuk melakukan penambahan dan penghapusan data itemnya. Data dari tab ini digunakan untuk melakukan penyaringan URL *request* dari *client* sebelum masuk ke pemrosesan lebih lanjut.

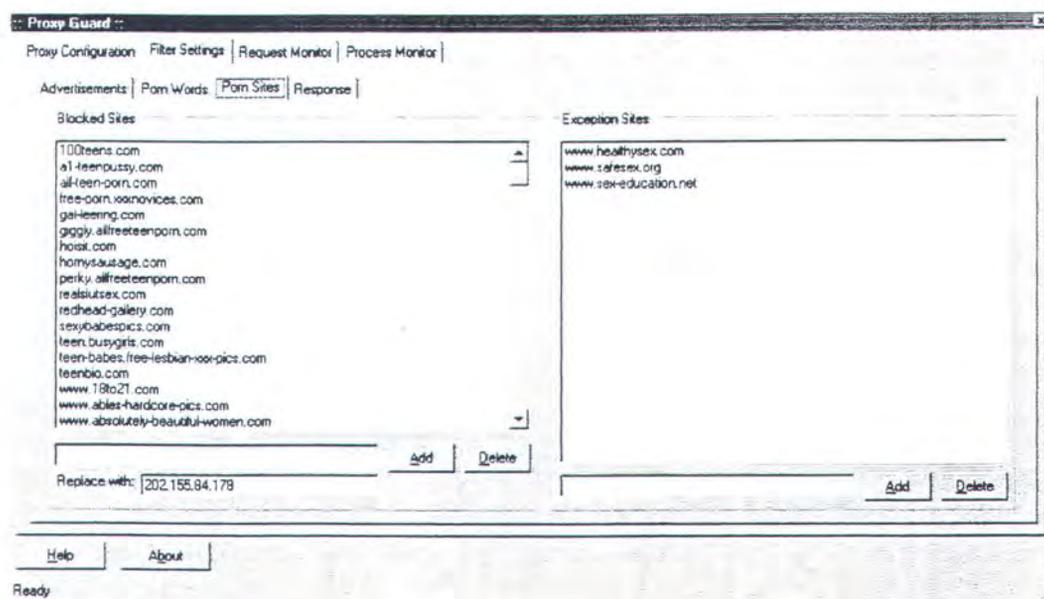
Tab Filter Settings - Porn Words



Gambar 3.7 Tampilan tab Filter Settings – Porn Words

Tab Porn Words memegang peran penting dalam penyaringan isi halaman sebuah file HTML. Komponen utama dalam tab ini adalah kontrol List Box yang berisikan *rule* untuk melakukan penyaringan. *Rule* terdiri atas lima bagian, yaitu bagian kata kunci pertama, bagian bobot kata kunci pertama, bagian tanda, bagian kata kunci kedua dan bagian bobot kata kunci kedua. *Rule* yang sedang dipilih bisa diedit isinya dengan menekan tombol Edit. Penambahan *rule* baru dapat dilakukan dengan menekan tombol Add setelah sebelumnya sudah mengisi data di text box yang tersedia.

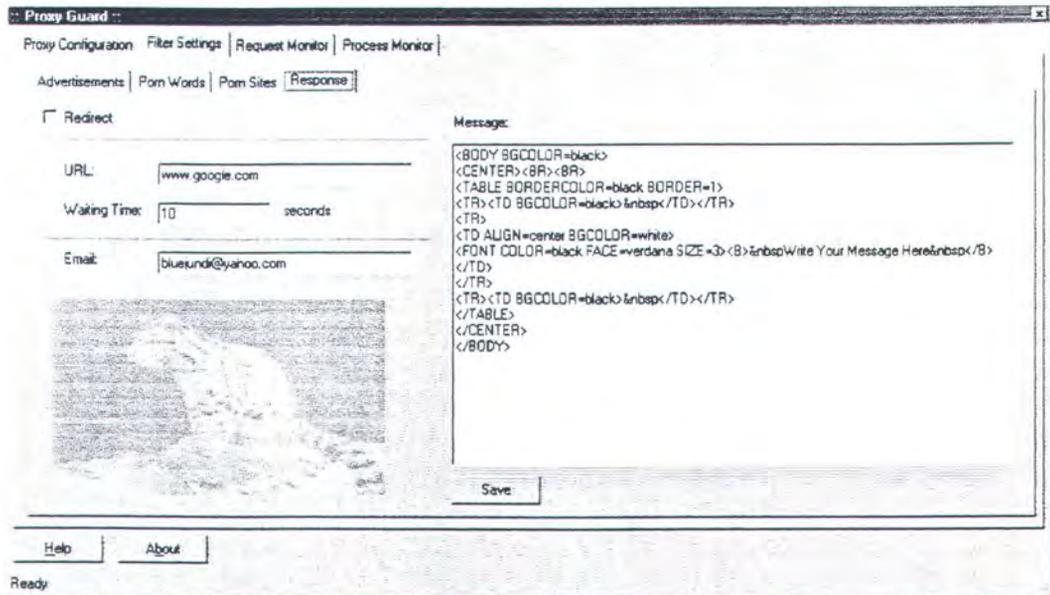
Tab Filter Settings – Porn Sites



Gambar 3.8 Tampilan tab Filter Settings – Porn Sites

Tab Porn Sites hamper sama isinya dengan tab Advertisement. Tab ini berisikan dua buah kotak kelompok yang masing-masing dilengkapi dengan fasilitas untuk melakukan penambahan dan penghapusan data itemnya. Satu kelompok berisikan daftar alamat situs-situs porno. Di kelompok yang lain berisikan alamat-alamat situs perkecualiannya. Sama seperti halnya tab Advertisement, komponen utama yang dipakai dalam tab ini adalah kontrol List Box. Di bagian bawah tab ini terdapat text box input untuk mengisi alamat URL yang akan dijadikan target *redirect* oleh aplikasi ketika penyaringan menghasilkan nilai positif.

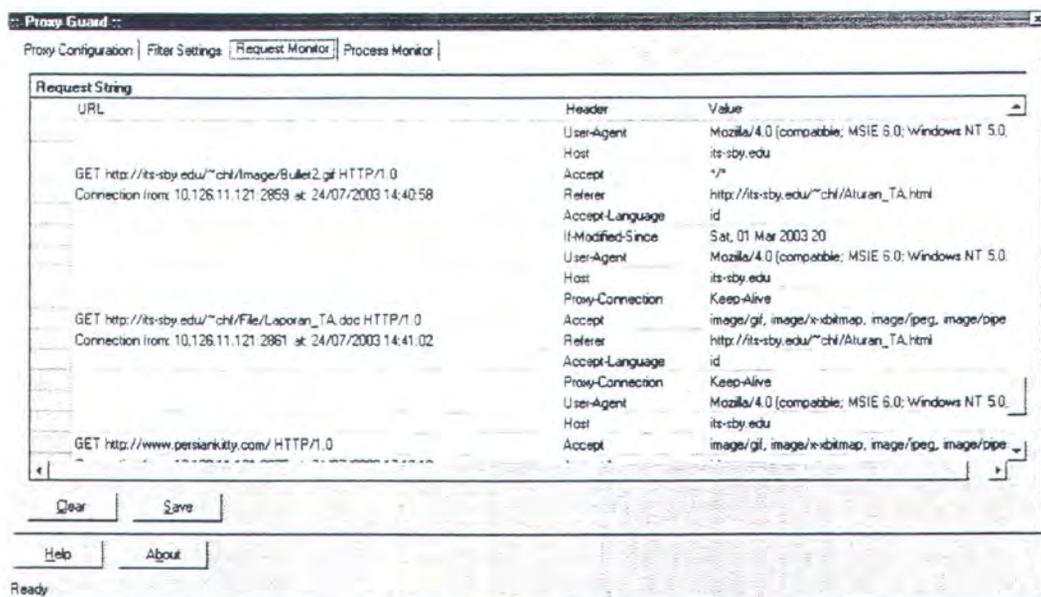
Tab Filter Settings – Response



Gambar 3.9 Tampilan tab Filter Settings - Response

Tab Response juga merupakan sarana bantu bagi pemakai untuk melakukan administrasi *proxy server*. Di dalam tab ini bisa diatur masalah URL untuk *redirect* dan waktu tungguanya. Dapat pula mengatur alamat email yang bisa dihubungi oleh *client* jika ada komplain dan sebagainya. Komponen Rich Text Box dipakai untuk melakukan *editing* pesan yang akan dikirimkan ke *client* jika terjadi pelanggaran akses. Teks yang dituliskan berformat HTML. Perubahan teks bisa disimpan ke file dengan menekan tombol Save.

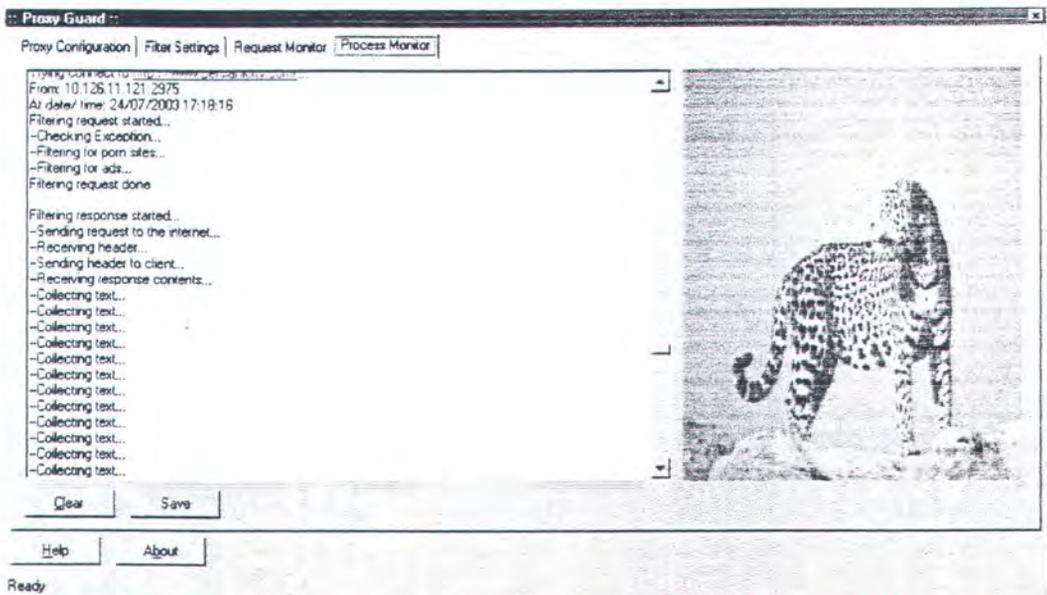
Tab Request Monitor



Gambar 3.10 Tampilan tab Request Monitor

Tab Request Monitor merupakan sarana bantu bagi pemakai untuk memantau dan mencatat *request client* yang masuk ke *proxy*. Komponen yang dipakai adalah Data Grid yang terhubung dengan sebuah Data Set. Data yang ditampilkan di grid berisikan *string request* yang sudah dipotong-potong beserta informasi tambahan berupa lokasi asal *client* dan waktu terjadinya *request*. *Update* isi grid dilakukan secara waktu nyata setiap kali ada *request* dari *client*. Grid yang sudah terlalu penuh bisa dihapus isinya dengan menekan tombol Clear. Namun jika dirasa perlu untuk menyimpannya ke dalam sebuah file teks, maka bisa ditekan tombol Save terlebih dahulu.

Tab Process Monitor



Gambar 3.11 Tampilan tab Process Monitor

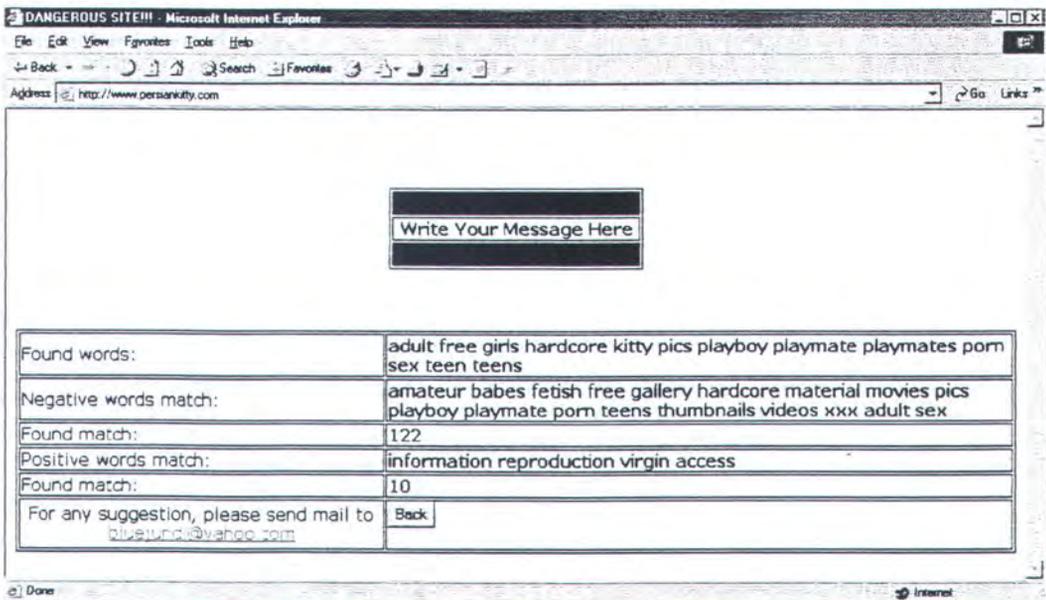
Tab Process Monitor juga merupakan sarana bantu untuk pemakai aplikasi Proxy Guard. Dengan tab ini pemakai bisa mengamati proses apa saja yang sedang berjalan di latar yang tidak tampak di permukaan. Komponen yang dipakai untuk menyimpan teks tersebut adalah kontrol Rich Text Box. Teks yang ditampilkan merupakan pesan yang dikirimkan oleh *thread socket* yang sedang bekerja memproses *request client*. Dengan demikian seandainya terjadi kesalahan bisa langsung diketahui dimana letaknya dalam proses penyaringan. Jika dirasa sudah terlalu penuh isinya, maka teks bisa dihapus dengan cara menekan tombol Clear. Namun jika dirasa perlu menyimpannya terlebih dahulu maka bisa ditekan tombol Save.

Tampilan Output Penyaringan Request



Gambar 3.12 Tampilan output hasil penyaringan request

Tampilan Output Penyaringan Response



Gambar 3.13 Tampilan output hasil penyaringan response

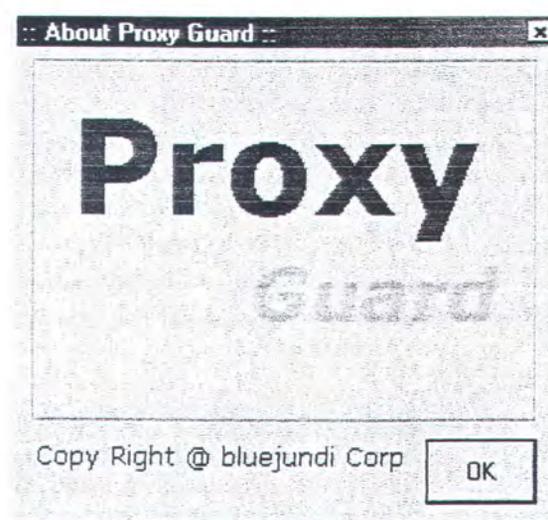
Gambar-gambar di atas merupakan contoh pesan yang akan ditampilkan di browser client jika terjadi pelanggaran rule. Berisikan pesan dari pemakai aplikasi Proxy Guard ditambahi dengan informasi tentang sebab-sebab request tidak bisa diproses dengan baik

Tampilan Window Help



Gambar 3.14 Tampilan window Help

Tampilan Windows About



Gambar 3.15 Tampilan window About

3.3.2. Proses Penyaringan

Berikut ini adalah algoritma penyaringan yang dilakukan oleh aplikasi Proxy Guard. Secara umum penyaringan dilakukan dua tahap, yaitu penyaringan *request* dilanjutkan dengan penyaringan isi halaman.

```

receive_request();
filter_request()
if (match==true)
    stop_request();
    redirect();
else
    send_request();
    receive_response();
    if (content!=text)
        send_response();
    elseif (content==text)
        extract_text();
        matching_with_the_rule();
        if (match!=true)
            send_response();
        elseif (match==true)
            rewrite_response();
            send_response();

```

3.3.2.1. Pengaktifan Thread Aplikasi

Pengaktifan *thread* aplikasi dilakukan ketika *event click* terjadi pada tombol Start. Dipilih objek *thread* karena aplikasi Proxy Guard membutuhkan proses latar yaitu menunggu *request* yang masuk. Jika ini dilakukan tanpa *thread* maka antarmuka yang ada bisa menjadi tidak responsif terhadap *event* yang dilakukan pemakai.

```

private void btnStart_Click(object sender, System.EventArgs e)
{
    if (btnStart.Enabled == true)
    {
        m_EventStopThread.Reset();
        m_EventThreadStopped.Reset();
    }
}

```

```

        TTWP = new Thread(new
ThreadStart(this.WorkerThreadFunction));
        TTWP.Name = "Worker Thread Sample";
        TTWP.Start();
        Console.WriteLine("Thread running");
        btnStart.Enabled = false;
        btnStop.Enabled = true;
    }
}
private void WorkerThreadFunction()
{
    myWebProxy myWP;
    myWP = new myWebProxy();
    myWP.run();
}

```

3.3.2.2. Pembuatan Thread Socket

Aplikasi Proxy Guard memerlukan proses anak yang berjalan di latar untuk menangani *request* dari *client*. Maka dari itu solusi pemrograman yang diberikan adalah menggunakan *class thread* yang terhubung dengan *socket*.

```

public myWebProxy ()
{
    int port = int.Parse(new Proxy_Guard.FormMain().textBoxPort.Text);
    try
    {
        TcpListener serversocket = new TcpListener(port);
        Console.WriteLine("Listening on port " + port);
        serversocket.Start();
        while (true)
        {
            Socket socket = serversocket.AcceptSocket();
            myWebProxy webproxy=new myWebProxy(socket);
            t =new Thread(new ThreadStart(webproxy.run));
            t.Start();
        }
    }
    catch (IOException ioe)
    { Console.WriteLine(ioe.ToString()); }
}

```



3.3.2.3. Penyaringan Request

Penyaringan *request* dimulai dengan proses pengambilan *substring* dari URL string *request* yang berisikan informasi nama *domain*. Nama *domain* ini nantinya akan dibandingkan dengan koleksi nama *domain* di daftar hitam.

```
int i1 = strRequest.IndexOf(' ');
int i2 = strRequest.IndexOf(' ', i1+1);
if (i1 == -1 || i2 == -1)
{
    throw new Exception("Data Recieved Is Not In Correct
Format");
}
string strUrl = strRequest.Substring(i1 + 1, i2-i1).Trim();
int i3 = strUrl.IndexOf("/", 0);
int i4 = strUrl.IndexOf("/", i3+2);
String host = strUrl.Substring(i3+2, i4-i3-2);
```

Contohnya jika ada URL dari *request string* seperti ini: <http://www.google.com/intl/id/about.html> maka nama domain yang didapatkan adalah www.google.com.

Jika terjadi kecocokan antara nama *domain* dengan salah satu alamat situs yang terdapat di dalam daftar hitam, maka pemrosesan *request* segera dihentikan. Kemudian *request client* diberi *response* berupa *redirect* ke alamat yang berbeda dengan yang seharusnya dituju. Selanjutnya koneksi *socket* ditutup. Dan tugas *thread* selesai sampai di sini.

```
if ((blockedporn==true)|| (blockedadds==true))
{
    string strReponse = "HTTP/1.0 302 Redirect\r\n"
        + "Content-type: text/html\r\n"
        + "Location: C:\\Temp\\TA\\neo\\Proxy_Guard\\redirect.html"
        + "\r\n\r\n";
    byte[] bufferResLogin = ASCIIEncoding.ASCII.GetBytes(strReponse);
    this.sockRes.Send(bufferResLogin,bufferResLogin.Length,SocketFlags
.None);
```

```

this.sockRes.Close();
return;
}

```

Namun jika *request* lolos dari penyaringan nama *domain* maka dilanjutkan proses pengiriman *request* ke *server* sebenarnya di internet. Selanjutnya *socket* akan terus menerima *response* dari *server* internet sampai jumlah bytes yang diterima sama dengan nol. *Response* berupa teks akan ditampung sementara dulu sampai seluruh isinya diterima. Hal ini dilakukan untuk keperluan penyaringan berdasar isi halaman web. Sedangkan *response* berupa file bukan teks akan langsung diteruskan ke *client*.

```

while (bytes > 0) //RECEIVE ALL RESPONSE
{
    bytes = sockReq.Receive(RecvBytes);
    if (strhead.IndexOf("Content-Type: text/")>0)
    {
        strResp.Append(Encoding.ASCII.GetString(RecvBytes, 0,
bytes));
        sresp = strResp.ToString();
    }
    else //EXCEPT NON TEXT FILE
    {
        this.sockRes.Send(RecvBytes, bytes, SocketFlags.None);
    }
}

```

3.3.2.4. Penyaringan Response

Penyaringan *response* dari *server* di internet diawali dengan proses ekstraksi teks yang terdapat dalam sebuah halaman web. Proses ekstraksi ini menggunakan layanan fungsi *Regex* yang dimiliki oleh C#. Pola *string* yang akan dihilangkan adalah segala macam bentuk *script* yang ada. Kemudian segala yang berhubungan dengan *tag* HTML beserta seluruh isi yang ada di antara pasangan

tagnya digantikan dengan sebuah tanda spasi. Demikian halnya dengan semua karakter spasi, karakter ganti baris, karakter bukan alphabet maupun karakter angka diganti dengan sebuah tanda spasi. Tanda spasi ini nanti akan sangat berguna untuk menentukan jumlah kata yang terdapat di dalam sebuah halaman web.

```

ArrayList ar=new ArrayList();
ar.Add("<script[^>]*?>[\\s\\S]*?</script>");
ar.Add("<[/!]*?[^<][\\s\\S]*?>");
ar.Add("&[\\s\\S]*?;");
ar.Add("[\\r\\n]");
ar.Add("[\\W]");
ar.Add("[\\d]+");
ar.Add("[ ]+");
ArrayList arrep=new ArrayList();
arrep.Add(""); arrep.Add(" ");
arrep.Add(" "); arrep.Add(" ");
arrep.Add(" "); arrep.Add(" ");
arrep.Add(" ");

string strout=sresp;
int i, j;
Regex rgx;
for (i=0;i<ar.Count;i++)
{
    rgx = new Regex(ar[i].ToString());
    strout = rgx.Replace(strout,arrep[i].ToString());
}

```

Setelah diperoleh teksnya maka dilakukan proses penghitungan frekuensi ditemukannya kata kunci pertama. Jika perbandingan jumlah antara kata kunci pertama dengan jumlah total kata yang ada lebih besar dari angka batas yang terdapat pada rule maka proses penghitungan frekuensi akan dihentikan. Kemudian proses akan dilanjutkan untuk penghitungan frekuensi kata kunci kedua.

```

if (string.Compare(word1, splits[j])==0)
{
    found++;
}

```

```

r = (float)found/word_count;
if (r>rf)
{
    go = true; break;
}
else
{
    go = false;
}
}

```

```

for (j=1; j<splits.Length; j++)
{
    if (string.Compare(word2, splits[j])==0)
    {
        found++;
    }
}

```

Jika dari hasil pencarian ditemukan frekuensi kata kunci kedua yang melebihi angka batasnya maka dilakukan pengiriman *response* pengganti kepada *client*. *Response* ini berisikan informasi mengapa *response* yang seharusnya diterima tidak bisa dikirimkan sesuai aslinya.

```

if (found>=founds)
{
    string email = (string)fm.Invoke(fm.m_DelegateGetTB, new
Object[] {fm.textBox9});
    string body = (string)fm.Invoke(fm.m_DelegateGetRTB, new
Object[] {fm.richTextBox2});
    sresp = this.build_header()
+
    "<body bgcolor=white><Script
Language=\"JavaScript\">alert(\"WARNING! You try to access porn
site!\");</Script>"
+ "<font color=black face=verdana size=5><center><b>"
+ body + "</b><br><br>"
+ "<table bordercolor=black border=1>"
+ "<tr><td><font color=black face=verdana size=3>Found
word:</td><td><b><font color=black face=verdana size=3>" + word1 +
"</b></font></td></tr>"
+ "<tr><td><font color=black face=verdana size=3>With
frequency at least:</td><td><b><font color=black face=verdana
size=3>" + founds.ToString() + "</b></font></td></tr>"
+ "<tr><td><font color=black face=verdana size=3>Followed
with word:</td><td><b><font color=black face=verdana size=3>" +
word2 + "</b></font></td></tr>"

```

```

+ "<tr><td><font color=black face=verdana size=3>With
frequency at least:</td><td><b><font color=black face=verdana
size=3>" + found.ToString() + "</b></font></td></tr>"
+ "<tr><td align=center bgcolor=white><font color=black
face=verdana size=3>For any suggestion, please send mail to "
+ "<a href=\"mailto:" + email + "\">" + email + "</td>"
+ "<td><form><input type=button name=back value=Back
onClick=\"history.go(-1)\"></form></td></tr>"
+ "</table>"
+ "</center></body></html>";
this.sockRes.Send(ASCIIEncoding.ASCII.GetBytes(sresp));

```

Namun jika hasil pencarian ternyata menemukan jumlah yang lebih sedikit dari pada angka batasnya maka *response* yang selama proses penyaringan ini ditahan akan segera dikirimkan ke *client*. Kemudian *socket* yang menghubungkan ditutup.

```

else
{this.sockRes.Send(ASCIIEncoding.ASCII.GetBytes(sresp));}

```



BAB IV
IMPLEMENTASI DAN UJI COBA

BAB IV IMPLEMENTASI DAN UJI COBA

4.1. Implementasi

Aplikasi Proxy Guard diimplementasikan pada sistem komputer dengan spesifikasi sebagai berikut:

<i>Sistem operasi</i>	: <i>Microsoft Windows 2000 Service Pack 4</i>
<i>Jenis komputer</i>	: <i>HP Vectra VE</i>
<i>Prosesor</i>	: <i>Pentium II 450 MHz</i>
<i>RAM</i>	: <i>359.988 KB</i>
<i>Hard Disk</i>	: <i>4.3 GB</i>
<i>LAN card</i>	: <i>3Com EtherLink XL 10/100 PCI</i>
<i>Parent proxy server</i>	: <i>202.155.84.180:8080; Squid/2.4 STABLE3</i>

4.2. Skenario Uji Coba

Aplikasi Proxy Guard diujicobakan untuk menyaring alamat-alamat situs yang dihasilkan dari pencarian menggunakan *search engine* Google dan Yahoo. Skenario uji coba dibagi menjadi tiga kategori, yaitu kategori situs yang terindikasi sebagai situs porno, kategori situs yang terindikasi sebagai situs bukan porno, dan kategori situs yang terindikasi netral.

Kategorisasi ini diperoleh berdasarkan pencarian melalui *search engine* dengan memasukkan kata kunci yang sesuai. Untuk kategori porno dimasukkan kata kunci “*sex*” “*porn*” dan “*xxx*”. Untuk kategori non porno dimasukkan kata kunci “*sex*” “*education*” dan “*health*”. Sedang untuk situs yang netral digunakan

kata kunci “sex” saja. Jadi semua *link* yang ditemukan belum pernah dipakai sebagai percobaan oleh aplikasi Proxy Guard selama masa pembuatan.

Selain itu juga akan dilakukan uji coba untuk mengetahui pengaruh penyaringan terhadap lamanya waktu pemrosesan *request* dari *client* sampai selesai, yaitu sampai dikirimkannya semua *response* dari *server* di internet ke *client*. Alamat-alamat situs yang digunakan sebagai sample adalah yang bisa dikelompokkan sebagai situs dominan berisi teks atau situs yang dominan non-teks.

4.2.1. Kemampuan Penyaringan

Uji coba ini dilakukan dalam kondisi hanya ada satu jendela browser yang aktif serta dalam kondisi sistem operasi sedang memiliki beberapa aplikasi aktif yang sama. Dengan demikian diharapkan ada kestabilan di internal sistem komputer. Hasil uji coba kemampuan penyaringan ditunjukkan oleh tabel-tabel sebagai berikut:

Tabel 4.1 Hasil uji coba untuk situs porno

Nama Situs	Hasil
www.persiankitty.com	Sukses diblok
www.bobs-free-sex-porn-pics.com	Sukses diblok
www.free-xxx-porn-pics-gallery.com	Sukses diblok
www.link-o-rama.com	Sukses diblok
www.super-porno.com	Sukses diblok
www.sublimedirectory.com	Sukses diblok
www.easylife.org/xxx/	Sukses diblok
www.xxxporntastic.com	Sukses diblok

www.freesitexxx.com	Sukses diblok
www.larrys-free-sex-porno-gallery.com	Sukses diblok
www.hoes.com	Sukses diblok
www.sexy nexus.com	Sukses diblok
www.pomoxxxgals.com	Sukses diblok
www.ambers-free-xxx-sex-pictures.com	Sukses diblok
www.private.com	Sukses diblok

Tabel 4.2 Hasil uji coba untuk situs bukan porno

Nama Situs	Hasil
www.sieccan.org	Lolos
news.bbc.co.uk/1/hi/health/2336643.stm	Lolos
www.atsdr.cdc.gov/glossary.html	Lolos
www.co.larimer.co.us/health/ed/reproductive.asp	Lolos
www.kalcounty.com/hsd/healthsafe.htm	Lolos
www.city.sendai.jp/kenkou/kodomo/sukoyaka/05-e.html	Lolos
www.electgop.net/vt/baralforhouse/a107.html	Lolos
dir.123india.com/health_and_medicine/resources/	Lolos
www.avert.org/sexedu.htm	Lolos
www.jocoks.com/pubhealth/healthed.htm	Lolos
www.amherst.edu/~healthed/	Lolos
www.unescobkk.org/ips/som/som-2002-01.htm	Lolos
www.etaiwannews.com/Health/2002/11/21/1037870872.htm	Lolos

www.sundayherald.com/35286	Lolos
www.mofa.go.jp/policy/human/women_rep5/part2_12.html	Lolos

Tabel 4.3 Hasil uji coba situs netral

Nama Situs	Hasil	Kategori
www.sfsi.org	Lolos	Non porno
www.sexuality.org	Lolos	Non porno
www.itsyoursexlife.com	Lolos	Non porno
www.allaboutsex.org	Lolos	Non porno
www.sex-pistols.net	Lolos	Non porno
www.salon.com/sex/	Sukses diblok	Porno
www.sexaa.org	Lolos	Non porno
www.sxetc.org	Lolos	Non porno
www.altsex.org	Lolos	Non porno
www.safersex.org	Lolos	Non porno
www.world-sex-news.com	Sukses diblok	Portal ke situs porno
www.sexthermometer.com	Sukses diblok	Non porno
www.positive.org/JustSayYes/safesex.html	Lolos	Non porno
www.sexrespect.com	Lolos	Non porno
www.luckymojo.com/sacredsex.html	Lolos	Non porno

Uji Coba Batas Rasio Penguat/Pelemah

Tabel 4.4 Hasil uji coba angka batas rasio

Nama situs	R = 20	R = 40	R = 60	R = 80
www.persiankitty.com	Blocked	Passed	Passed	Passed
www.bobs-free-sex-porn-pics.com	Blocked	Blocked	Blocked	Passed
www.free-xxx-porn-pics-gallery.com	Blocked	Blocked	Blocked	Blocked
www.link-o-rama.com	Blocked	Blocked	Blocked	Blocked
www.super-porno.com	Blocked	Blocked	Passed	Passed

4.2.2. Perbandingan Waktu

Sudah ada hipotesa di awal bahwasanya akan terjadi pelambatan proses sebagai dampak dari penyaringan. Karena jelas terjadi proses tambahan dibandingkan proses *defaultnya*. Terkecuali jika memang *request* sudah terblok sebelum diproses *responsnya* dikarenakan termasuk dalam daftar hitam situs terlarang.

Uji coba ini dilakukan untuk mengetahui sebesar apa pengaruh proses penyaringan isi teks file html terhadap waktu pemrosesan *request client* sampai selesai. Waktu yang dipakai oleh aplikasi Proxy Guard dibandingkan antara yang melalui proses penyaringan isi dengan yang tidak melakukan penyaringan isi. Target situs dikelompokkan menjadi dua yaitu situs yang sudah dikenal minim isi teksnya dan situs yang relatif banyak berisikan teks.

Daftar situs yang dijadikan target percobaan serta hasilnya adalah sebagai berikut:

Tabel 4.5 Hasil uji coba situs dominan bukan teks

Nama Situs	Tanpa filter	Dengan filter	Rasio
www.google.com	4	7	1.75
mail.yahoo.com	19	22	1.16
mail.telkom.net	60	163	2.72
mail.erasuslim.com	61	132	2.16

Tabel 4.6 Hasil uji coba situs dominant teks

Nama situs	Tanpa filter	Dengan filter	Rasio
www.yahoo.com	31	68	2.19
www.erasuslim.com	120	231	1.92
www.detik.com	314	321	1.02
www.republika .co.id	39	119	3.05
www.keadilan.or.id	84	123	1.46
www.peka.or.id	48	121	2.52

4.3. Evaluasi

Hasil uji coba di atas menunjukkan bahwa 100% sample situs porno bisa diblok, dan 100% sample situs non porno bisa ditampilkan ke *client*. Namun untuk kategori situs yang diindikasikan netral ternyata terjadi satu kasus *overblocking*, dimana situs yang seharusnya aman untuk diteruskan ke *client* ternyata juga ikut terblokir oleh aplikasi.

Uji coba batas rasio untuk situs porno memperlihatkan bahwa masing-masing situs mempunyai nilai batas yang berbeda-beda. Untuk amannya maka dipilihlah nilai batas yang paling kecil sehingga bisa memblokir semua situs porno mulai dari batas terendah dan selebihnya.

Dari hasil uji coba perbandingan waktu jelas terlihat adanya pelipatgandaan waktu pemrosesan *request* dari *client* yang melalui *filter* isi halaman. Pelipatgandaan tersebut berkisar antara 1.02 kali sampai 3.05 kali lebih lama dibandingkan pemrosesan tanpa melalui penyaringan.

Dari data di atas pula dapat disimpulkan bahwa semakin banyak teks yang terdapat di dalam sebuah halaman web maka waktu pemrosesannya akan juga semakin lama. Hal ini dibuktikan dengan angka rasio di tabel daftar situs minim teks mengalami pelipatgandaan waktu proses yang lebih kecil dibandingkan dengan angka rasio situs yang relatif banyak mengandung teks.



BAB V
KESIMPULAN DAN SARAN

BAB V KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah dilakukan uji coba maka dapat diambil kesimpulan bahwa:

1. Dengan menggunakan *class socket* dan *multithreading*, aplikasi Proxy Guard mampu menjalankan perannya sebagai *proxy server* dalam menangani *request* dari *client* ke *server* di internet dengan baik.
2. Aplikasi Proxy Guard juga mampu melakukan penghalangan terhadap *request* yang bermaksud mengakses ke alamat situs terlarang. Teknik penghalangan ini dilakukan dengan jalan mengambil informasi nama *domain/ host* dari URL yang terdapat dalam *request string* dari *client*. Hasil perbandingan antara nama *domain* dengan koleksi nama *domain* terlarang akan menentukan boleh tidaknya *request* diteruskan.
3. Aplikasi ini berhasil pula melakukan kategorisasi konteks sebuah halaman web apakah termasuk situs porno ataukah bukan berdasarkan teksnya, dengan memakai metode penentuan konteks. Metode ini menggunakan *rule* berisi satu pasang kata kunci disertai bobot masing-masing dan sebuah karakter tanda pengaruh yang berfungsi menguatkan atau melemahkan konteks halaman internet yang sedang diproses.
4. Aplikasi Proxy Guard memberikan kemudahan dalam hal pengoperasiannya. Beberapa menu *administrating proxy server* bisa dilakukan oleh aplikasi ini secara waktu nyata tanpa harus melakukan

restart sistem. Kemampuan ini bisa dilakukan dengan menggunakan bantuan *class delegate* dari koleksi C# yang memungkinkan antarmuka bisa berkomunikasi dengan sebuah *thread* yang sedang berjalan tanpa harus mengganggu koresponsifannya.

5.2. Saran

Sebagaimana telah dijelaskan pada BAB I Dasar Teori tentang Internet Penyingkiran, bahwasanya dalam pengembangan sebuah aplikasi/ sistem untuk memfilter internet adalah sulit untuk mendapatkan validitas keberhasilan yang mencapai angka 100%. Demikian halnya dengan aplikasi Proxy Guard ini. Tetap saja masih ditemukan celah-celah yang bisa lolos dari penyingkiran. Salah satu hal penyebab yang utama adalah karena dunia internet merupakan dunia yang teramat luas, plural dan senantiasa terus tumbuh dan berkembang. Dan di sisi yang lain adalah keterbatasan kemampuan aplikasi dalam melakukan penyingkiran.

Untuk itu pengembangan teknik penyingkiran internet perlu terus dilanjutkan untuk mendapatkan teknik yang paling optimum. Sebagai saran perbaikan ke depan adalah:

1. Meningkatkan kemampuan daftar *rule* yang ada sehingga bisa semakin memperkecil peluang lolosnya akses ke situs terlarang. Kemampuan yang perlu diperbaiki tersebut adalah baik yang berhubungan dengan daftar koleksi kata-kata kuncinya maupun besar kecil angka batas yang menjadi parameternya.



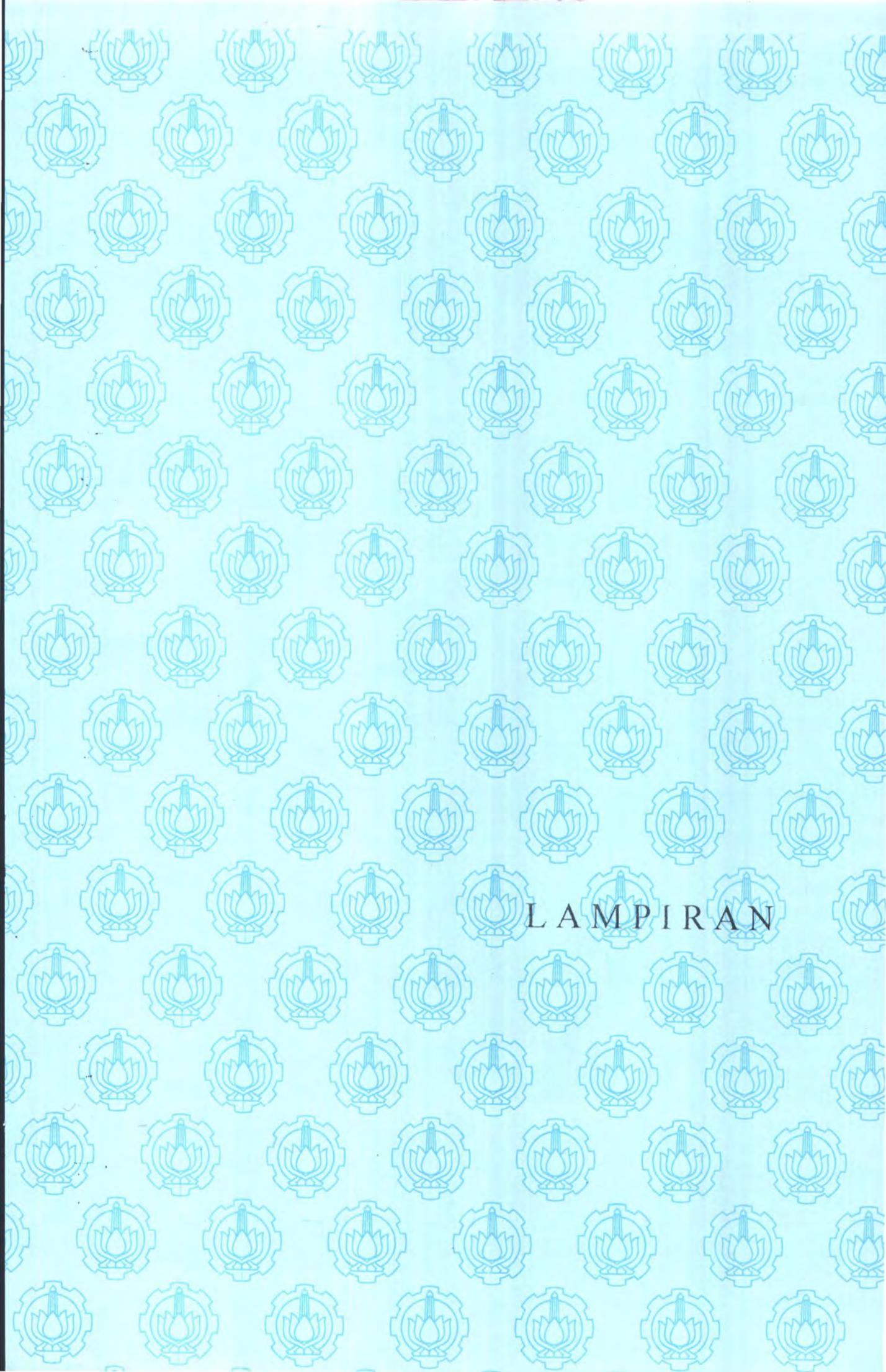
2. Melakukan optimasi algoritma pencarian dan pengurutan rule-nya sehingga rule yang ada mampu bekerja dengan lebih cepat dan efisien.
3. Melengkapi fasilitas aplikasi Proxy Guard dengan menambahkan pilihan-pilihan setting yang lebih mendetail berkenaan dengan administrasi yang dibutuhkan oleh sebuah *proxy server*. Dengan demikian diharapkan kemudahan yang diberikan bagi pemakai akan menjadi lebih baik



DAFTAR PUSTAKA

DAFTAR PUSTAKA

- [ARI] Luotonen, Ari; "Web Proxy Server"; Prentice Hall; 1998
- [CSH] <http://www.csharpelp.com>
- [EDI] Purwono, Edi; <http://members.tripod.com/~tiponline/top1021.htm>
- [FE] <http://www.fedu.uec.ac.jp/>
- [MSD] Microsoft Service Digital Network 2002
- [PG] Paul Greenfield, Peter Rickwood, Huu Cuong Tran; "Effectiveness of Internet Filtering Software Products"; CSIRO Mathematical and Information Sciences; September 2001



LAMPIRAN

LAMPIRAN

DAFTAR RULE UNTUK PENYARINGAN

adult	0.001	-	aids	1	adult	0.001	-	herpes	1
adult	0.001	-	baby	1	adult	0.001	-	hiv	1
adult	0.001	-	biology	1	adult	0.001	-	hospital	1
adult	0.001	-	birth	1	adult	0.001	-	hygiene	1
adult	0.001	-	bleed	1	adult	0.001	-	information	1
adult	0.001	-	catheter	1	adult	0.001	-	laws	1
adult	0.001	-	ensorship	1	adult	0.001	-	marriage	1
adult	0.001	-	chromosome	1	adult	0.001	-	medicine	1
adult	0.001	-	clinic	1	adult	0.001	-	menopause	1
adult	0.001	-	communication	1	adult	0.001	-	menstrual	1
adult	0.001	-	condom	1	adult	0.001	-	milk	1
adult	0.001	-	consultation	1	adult	0.001	-	pregnant	1
adult	0.001	-	contraception	1	adult	0.001	-	prenatal	1
adult	0.001	-	counseling	1	adult	0.001	-	problem	1
adult	0.001	-	crime	1	adult	0.001	-	recovery	1
adult	0.001	-	diabetes	1	adult	0.001	-	rehabilitation	1
adult	0.001	-	dictionary	1	adult	0.001	-	reproduction	1
adult	0.001	-	discussion	1	adult	0.001	-	research	1
adult	0.001	-	disease	1	adult	0.001	-	safe	1
adult	0.001	-	education	1	adult	0.001	-	scholarship	1
adult	0.001	-	fetal	1	adult	0.001	-	social	1
adult	0.001	-	guide	1	adult	0.001	-	surgery	1
adult	0.001	-	health	1	adult	0.001	-	symptom	1
adult	0.001	-	hermaphrodit	1	adult	0.001	-	testosteron	1
					adult	0.001	-	therapy	1

adult	0.001	-	tumor	1	adult	0.001	+	fetish	1
adult	0.001	-	virgin	1	adult	0.001	+	free	1
adult	0.001	-	virginity	1	adult	0.001	+	fuck	1
adult	0.001	-	virus	1	adult	0.001	+	fucking	1
adult	0.001	-	vitamins	1	adult	0.001	+	gallery	1
adult	0.001	+	amateur	1	adult	0.001	+	gay	1
adult	0.001	+	anal	1	adult	0.001	+	hardcore	1
adult	0.001	+	asian	1	adult	0.001	+	hentai	1
adult	0.001	+	ass	1	adult	0.001	+	hot	1
adult	0.001	+	babes	1	adult	0.001	+	kinky	1
adult	0.001	+	bitch	1	adult	0.001	+	kneeling	1
adult	0.001	+	bitches	1	adult	0.001	+	lesbian	1
adult	0.001	+	blowjob	1	adult	0.001	+	lick	1
adult	0.001	+	bondage	1	adult	0.001	+	lingerie	1
adult	0.001	+	butts	1	adult	0.001	+	lust	1
adult	0.001	+	celeb	1	adult	0.001	+	masturbate	1
adult	0.001	+	celebrity	1	adult	0.001	+	masturbation	1
adult	0.001	+	celebs	1	adult	0.001	+	material	1
adult	0.001	+	clit	1	adult	0.001	+	moan	1
adult	0.001	+	clits	1	adult	0.001	+	movies	1
adult	0.001	+	cocks	1	adult	0.001	+	mpegs	1
adult	0.001	+	cum	1	adult	0.001	+	naked	1
adult	0.001	+	cums	1	adult	0.001	+	naughty	1
adult	0.001	+	cunt	1	adult	0.001	+	nude	1
adult	0.001	+	cyber	1	adult	0.001	+	nympho	1
adult	0.001	+	dick	1	adult	0.001	+	oral	1
adult	0.001	+	dicks	1	adult	0.001	+	orgasm	1
adult	0.001	+	dildo	1	adult	0.001	+	orgy	1
adult	0.001	+	erotic	1	adult	0.001	+	pics	1
adult	0.001	+	explicit	1	adult	0.001	+	pictures	1
adult	0.001	+	fellatio	1	adult	0.001	+	playboy	1

adult	0.001	+	playmate	1	celeb	0.001	+	xxx	1
adult	0.001	+	porn	1	celebrity	0.001	+	xxx	1
adult	0.001	+	pornography	1	cerita	0.001	+	seru	1
adult	0.001	+	pussy	1	clitoris	0.001	-	health	1
adult	0.001	+	seductive	1	clitoris	0.001	-	vulva	1
adult	0.001	+	shemale	1	clitoris	0.001	+	xxx	1
adult	0.001	+	show	1	clits	0.001	+	xxx	1
adult	0.001	+	sigh	1	cocks	0.001	-	hen	1
adult	0.001	+	sluts	1	cocks	0.001	-	chicken	1
adult	0.001	+	strip	1	cocks	0.001	-	bird	1
adult	0.001	+	striptease	1	cocks	0.001	+	xxx	1
adult	0.001	+	suck	1	cocks	0.001	+	nude	1
adult	0.001	+	teens	1	cocks	0.001	+	naked	1
adult	0.001	+	thumbnails	1	cocks	0.001	+	tits	1
adult	0.001	+	tits	1	cyber	0.001	-	friend	1
adult	0.001	+	underwear	1	cyber	0.001	-	law	1
adult	0.001	+	videos	1	cyber	0.001	+	porn	1
adult	0.001	+	voyeur	1	cyber	0.001	+	sex	1
adult	0.001	+	whore	1	cyber	0.001	+	xxx	1
adult	0.001	+	xxx	1	dewasa	0.001	-	diskusi	1
anal	0.001	+	xxx	1	dewasa	0.001	-	dokter	1
breast	0.001	-	cancer	1	dewasa	0.001	-	ekonomi	1
breast	0.001	-	surgery	1	dewasa	0.001	-	hormon	1
breast	0.001	-	tumor	1	dewasa	0.001	-	hukum	1
breast	0.001	+	xxx	1	dewasa	0.001	-	informasi	1
breast	0.001	+	nude	1	dewasa	0.001	-	kejahatan	1
breast	0.001	+	naked	1	dewasa	0.001	-	kelainan	1
bust	0.001	+	xxx	1	dewasa	0.001	-	kelamin	1
bust	0.001	-	tissue	1	dewasa	0.001	-	keluhan	1
butt	0.001	+	xxx	1	dewasa	0.001	-	kesehatan	1
butt	0.001	-	tissue	1	dewasa	0.001	-	klinik	1

dewasa0.001	-	konsultasi	1	dirty	0.001	-	health	1
dewasa0.001	-	nikah	1	dirty	0.001	+	sex	1
dewasa0.001	-	obat	1	dirty	0.001	+	xxx	1
dewasa0.001	-	pendidikan	1	ebony	0.001	-	wood	1
dewasa0.001	-	penyakit	1	ebony	0.001	+	xxx	1
dewasa0.001	-	penyimpangan	1	fantasy	0.001	-	sleep	1
dewasa0.001	-	psikologi	1	fantasy	0.001	+	xxx	1
dewasa0.001	-	reproduksi	1	free	0.001	-	access	1
dewasa0.001	-	sakit	1	free	0.001	+	xxx	1
dewasa0.001	-	sehat	1	gay	0.001	-	social	1
dewasa0.001	-	seminar	1	gay	0.001	+	xxx	1
dewasa0.001	-	sosial	1	genital	0.001	-	disease	1
dewasa0.001	-	survei	1	genital	0.001	-	health	1
dewasa0.001	-	terapi	1	genital	0.001	+	xxx	1
dewasa0.001	-	zina	1	girls	0.001	+	parents	1
dewasa0.001	+	bokep	1	girls	0.001	+	xxx	1
dewasa0.001	+	bugil	1	hardcore	0.001	-		
dewasa0.001	+	cerita	1				construction	1
dewasa0.001	+	dewasa	1	hardcore	0.001	+	adult	1
dewasa0.001	+	foto	1	hardcore	0.001	+	xxx	1
dewasa0.001	+	gambar	1	horn	0.001	+	xxx	1
dewasa0.001	+	hentai	1	kitty	0.001	+	xxx	1
dewasa0.001	+	kamasutra	1	lesbian	0.001	+	social	1
dewasa0.001	+	masturbasi	1	lesbian	0.001	+	xxx	1
dewasa0.001	+	ngentot	1	masturbate	0.001	+	xxx	1
dewasa0.001	+	onani	1	masturbation	0.001	-		
dewasa0.001	+	orgasme	1				education	1
dewasa0.001	+	porno	1	naked	0.001	-	bath	1
dewasa0.001	+	telanjang	1	naked	0.001	+	xxx	1
dewasa0.001	+	vcd	1	nasty	0.001	+	xxx	1
dewasa0.001	+	wiro	1	naughty	0.001	-	child	1

naughty	0.001	+	xxx	1	playmate	0.001	+	playboy
nipple	0.001	+	health	1	1			
nipple	0.001	+	xxx	1	playmates	0.001	+	playboy
nympho	0.001	-	god	1	1			
nympho	0.001	+	xxx	1	porn	0.001	-	against 1
onani	0.001	-	consultation	1	porn	0.001	-	copyright 1
onani	0.001	+	xxx	1	porn	0.001	+	xxx 1
orgasm	0.001	-	health	1	prostitute	0.001	-	law 1
oral	0.001	-	medicine	1	prostitute	0.001	-	problem
oral	0.001	+	xxx	1	1			
orgasm	0.001	-	consultation	1	prostitute	0.001	+	xxx 1
orgasm	0.001	-	problem	1	pussy	0.001	+	xxx 1
orgasm	0.001	+	xxx	1	rectal	0.001	+	xxx 1
passion	0.001	-	goals	1	seks	0.001	-	agama 1
passion	0.001	+	xxx	1	seks	0.001	-	berita 1
pelacur	0.001	-	penanganan	1	seks	0.001	-	diskusi 1
pelacur	0.001	-	penangkapan	1	seks	0.001	-	dokter 1
pelacur	0.001	+	foto	1	seks	0.001	-	ekonomi 1
penis	0.001	-	disease	1	seks	0.001	-	hormon 1
penis	0.001	-	health	1	seks	0.001	-	hukum 1
penis	0.001	-	reproduction	1	seks	0.001	-	informasi 1
penis	0.001	+	bitch	1	seks	0.001	-	kejahatan 1
penis	0.001	+	fucks	1	seks	0.001	-	kelainan 1
penis	0.001	+	nude	1	seks	0.001	-	kelamin 1
penis	0.001	+	sucks	1	seks	0.001	-	keluhan 1
pics	0.001	+	xxx	1	seks	0.001	-	kesehatan 1
playboy	0.001	+			seks	0.001	-	klinik 1
playmate	1				seks	0.001	-	konsultasi 1
playboy	0.001	+	sex	1	seks	0.001	-	nikah 1
playboy	0.001	+	xxx	1	seks	0.001	-	obat 1
playgirl	0.001	+	xxx	1	seks	0.001	-	pendidikan 1

seks	0.001	-	penyakit	1	sex	0.001	-	aids	1
seks	0.001	-	penyimpangan	1	sex	0.001	-	baby	1
seks	0.001	-	psikologi	1	sex	0.001	-	biology	1
seks	0.001	-	reproduksi	1	sex	0.001	-	birth	1
seks	0.001	-	sakit	1	sex	0.001	-	bleed	1
seks	0.001	-	sehat	1	sex	0.001	-	catheter	1
seks	0.001	-	seminar	1	sex	0.001	-	copyright	1
seks	0.001	-	sosial	1	sex	0.001	-	chromosome	1
seks	0.001	-	survei	1	sex	0.001	-	clinic	1
seks	0.001	-	terapi	1	sex	0.001	-	communication	1
seks	0.001	-	zina	1					
seks	0.001	+	bokep	1	sex	0.001	-	condom	1
seks	0.001	+	bugil	1	sex	0.001	-	consultation	1
seks	0.001	+	cerita	1	sex	0.001	-	contraception	1
seks	0.001	+	dewasa	1	sex	0.001	-	counseling	1
seks	0.001	+	foto	1	sex	0.001	-	crime	1
seks	0.001	+	gambar	1	sex	0.001	-	diabetes	1
seks	0.001	+	hentai	1	sex	0.001	-	dictionary	1
seks	0.001	+	kamasutra	1	sex	0.001	-	discussion	1
seks	0.001	+	masturbasi	1	sex	0.001	-	disease	1
seks	0.001	+	ngentot	1	sex	0.001	-	doctor	1
seks	0.001	+	onani	1	sex	0.001	-	education	1
seks	0.001	+	orgasme	1	sex	0.001	-	equity	1
seks	0.001	+	porno	1	sex	0.001	-	estrogen	1
seks	0.001	+	telanjang	1	sex	0.001	-	fetal	1
seks	0.001	+	vcd	1	sex	0.001	-	fungal	1
seks	0.001	+	wiro	1	sex	0.001	-	gender	1
senggama	0.001	+	foto	1	sex	0.001	-	gene	1
senggama	0.001	+			sex	0.001	-	guide	1
			konsultasi	1	sex	0.001	-	health	1
sex	0.001	-	abuse	1	sex	0.001	-	hermaphrodit	1

sex	0.001	-	herpes	1	sex	0.001	-	social	1
sex	0.001	-	hiv	1	sex	0.001	-	surgery	1
sex	0.001	-	hospital	1	sex	0.001	-	symptom	1
sex	0.001	-	hygiene	1	sex	0.001	-	testosteron	1
sex	0.001	-	impotence	1	sex	0.001	-	therapy	1
sex	0.001	-	infection	1	sex	0.001	-	tissue	1
sex	0.001	-	information	1	sex	0.001	-	tumor	1
sex	0.001	-	irritation	1	sex	0.001	-	urethra	1
sex	0.001	-	labia	1	sex	0.001	-	urine	1
sex	0.001	-	laws	1	sex	0.001	-	uterine	1
sex	0.001	-	marriage	1	sex	0.001	-	uterus	1
sex	0.001	-	medicine	1	sex	0.001	-	vaginitis	1
sex	0.001	-	membran	1	sex	0.001	-	virgin	1
sex	0.001	-	menopause	1	sex	0.001	-	virginity	1
sex	0.001	-	menstrual	1	sex	0.001	-	virus	1
sex	0.001	-	milk	1	sex	0.001	-	vitamins	1
sex	0.001	-	offender	1	sex	0.001	-	vulva	1
sex	0.001	-	ovary	1	sex	0.001	+	adult	1
sex	0.001	-	pregnant	1	sex	0.001	+	amateur	1
sex	0.001	-	prenatal	1	sex	0.001	+	anal	1
sex	0.001	-	problem	1	sex	0.001	+	asian	1
sex	0.001	-	prostat	1	sex	0.001	+	ass	1
sex	0.001	-	puber	1	sex	0.001	+	babes	1
sex	0.001	-	recovery	1	sex	0.001	+	bitch	1
sex	0.001	-	rehabilitation	1	sex	0.001	+	bitches	1
sex	0.001	-	reproduction	1	sex	0.001	+	blowjob	1
sex	0.001	-	research	1	sex	0.001	+	bondage	1
sex	0.001	-	safe	1	sex	0.001	+	butts	1
sex	0.001	-	scholarship	1	sex	0.001	+	celeb	1
sex	0.001	-	scrotum	1	sex	0.001	+	celebrity	1
sex	0.001	-	semen	1	sex	0.001	+	celebs	1

sex	0.001	+	clit	1	sex	0.001	+	mpegs	1
sex	0.001	+	clits	1	sex	0.001	+	naked	1
sex	0.001	+	cocks	1	sex	0.001	+	naughty	1
sex	0.001	+	cum	1	sex	0.001	+	nude	1
sex	0.001	+	cums	1	sex	0.001	+	nympho	1
sex	0.001	+	cunt	1	sex	0.001	+	oral	1
sex	0.001	+	cyber	1	sex	0.001	+	orgasm	1
sex	0.001	+	dick	1	sex	0.001	+	orgy	1
sex	0.001	+	dicks	1	sex	0.001	+	pics	1
sex	0.001	+	dildo	1	sex	0.001	+	pictures	1
sex	0.001	+	erotic	1	sex	0.001	+	playboy	1
sex	0.001	+	fellatio	1	sex	0.001	+	playmate	1
sex	0.001	+	fetish	1	sex	0.001	+	porn	1
sex	0.001	+	free	1	sex	0.001	+	pornography	1
sex	0.001	+	fuck	1	sex	0.001	+	pussy	1
sex	0.001	+	fucking	1	sex	0.001	+	seductive	1
sex	0.001	+	gallery	1	sex	0.001	+	shemale	1
sex	0.001	+	gay	1	sex	0.001	+	show	1
sex	0.001	+	hardcore	1	sex	0.001	+	sigh	1
sex	0.001	+	hentai	1	sex	0.001	+	sluts	1
sex	0.001	+	hot	1	sex	0.001	+	strip	1
sex	0.001	+	kinky	1	sex	0.001	+	striptease	1
sex	0.001	+	kneeling	1	sex	0.001	+	suck	1
sex	0.001	+	lesbian	1	sex	0.001	+	teens	1
sex	0.001	+	lick	1	sex	0.001	+	thumbnails	1
sex	0.001	+	lingerie	1	sex	0.001	+	tits	1
sex	0.001	+	lust	1	sex	0.001	+	underwear	1
sex	0.001	+	masturbate	1	sex	0.001	+	videos	1
sex	0.001	+	masturbation	1	sex	0.001	+	voyeur	1
sex	0.001	+	moan	1	sex	0.001	+	whore	1
sex	0.001	+	movies	1	sex	0.001	+	xxx	1

shop	0.001	-	food	1	wet	0.001	-	water	1
shop	0.001	+	xxx	1	wet	0.001	+	xxx	1
sperm	0.001	-	fertilization	1	wild	0.001	-	life	1
sperm	0.001	-	ovum	1	wild	0.001	-	jungle	1
sperm	0.001	-	reproduction	1	wild	0.001	-	xxx	1
sperm	0.001	+	xxx	1					
stockings	0.001	-	dress	1					
stockings	0.001	+	clothes	1					
stockings	0.001	+	xxx	1					
strip	0.001	+	naked	1					
suck	0.001	-	baby	1					
suck	0.001	-	milk	1					
suck	0.001	-	nursing	1					
suck	0.001	+	xxx	1					
teen	0.001	-	education	1					
teen	0.001	-	health	1					
teen	0.001	-	information	1					
teen	0.001	-	orientation	1					
teen	0.001	-	pregnancy	1					
teen	0.001	-	puberty	1					
teen	0.001	-	youth	1					
teen	0.001	+	pics	1					
teen	0.001	+	xxx	1					
teens	0.001	+	nude	1					
telanjang	0.001	-	operasi	1					
telanjang	0.001	+	foto	1					
threesome	0.001	+	xxx	1					
tits	0.001	+	xxx	1					
vagina	0.001	+	nude	1					
wet	0.001	-	flood	1					
wet	0.001	-	rain	1					