



MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

**PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK
PENGAMBILAN INFORMASI SITUS BERITA SECARA
OTOMATIS MENGGUNAKAN *REGULAR EXPRESSION*
DENGAN PLATFORM MICROSOFT .NET**

TUGAS AKHIR



Disusun oleh :

ARIO KOESNAN

NRP. 5198 100 037

PERPUSTAKAAN ITS	
Tgl. Terima	19 - 8 - 2003
Terima Dari	H
No. Agenda Prp.	218878

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

2003

125/ITS/H/2004

**PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK
PENGAMBILAN INFORMASI SITUS BERITA SECARA
OTOMATIS MENGGUNAKAN *REGULAR EXPRESSION*
DENGAN PLATFORM MICROSOFT .NET**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer**

Pada

**Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya**

RSIf.
005.1
KoE.
P-J

2003.

Mengetahui / Menyetujui,

Dosen Pembimbing I

Dosen Pembimbing II

(Yudhi Purwananto, M.Kom)
NIP. 131 172 210

(Faizal Johan Atletiko, S.Kom)
NIP. 132 300 414

**SURABAYA
AGUSTUS, 2003**

ABSTRAK

Masyarakat belakangan ini banyak disuguhkan berbagai macam situs berita dengan berbagai macam kategori mulai dari yang thin content hingga rich content. Situs-situs tersebut umumnya tidak hanya menyajikan berita sebagai informasi utama, tetapi juga sebagai tempat untuk promosi, dalam hal ini penempatan banner-banner yang cukup banyak dijumpai khususnya pada situs-situs yang sudah banyak dikenal masyarakat seperti detik.com.

Situs-situs berita memiliki kesamaan satu dengan yang lainnya, dalam hal ini disebut pola (pattern). Untuk satu situs berita, umumnya bagian indeks dan direktorinya memiliki pola yang sama dalam struktur situsnya. Inilah yang umumnya disebut web content management.

Regular expression telah banyak dipakai untuk memanipulasi suatu string atau barisan karakter, mulai dari kata sampai paragraf. Dalam kasus ini, regular expression sangat efektif untuk dimanfaatkan dalam mengadopsi struktur pola suatu situs berita.

Web service yang menggunakan teknologi XML dan protokol HTTP dapat digunakan untuk sistem Client-Server. Dengan web service, Administrator dan Client dapat mengakses database untuk menjalankan fungsinya masing-masing. Teknologi Windows service bekerja sebagai background process menjalankan fungsinya sebagai pengambil content dari situs berita.

Kata kunci : Regular expression, Web service, Windows Service, Backgorund process

KATA PENGANTAR

Segala puji dan syukur semata ditujukan ke hadirat ALLAH SWT yang telah memberikan rahmat dan hidayah-Nya sehingga memungkinkan penulis untuk menyelesaikan Tugas Akhir yang berjudul “**PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK PENGAMBILAN INFORMASI SITUS BERITA SECARA OTOMATIS MENGGUNAKAN REGULAR EXPRESSION DENGAN PLATFORM MICROSOFT .NET**”.

Mata Kuliah Tugas Akhir yang memiliki beban sebesar 4 satuan kredit disusun dan diajukan sebagai salah satu syarat untuk menyelesaikan program strata satu (S-1) pada jurusan Teknik Informatika di Institut Teknologi Sepuluh Nopember Surabaya.

Dalam penyusunan Tugas Akhir ini, Penulis berusaha untuk menerapkan ilmu yang telah didapat selama menjalani perkuliahan dengan tidak terlepas dari petunjuk, bimbingan, bantuan, dan dukungan berbagai pihak.

Dengan tidak lupa akan kodratnya sebagai manusia, Penulis menyadari bahwa dalam karya Tugas Akhir ini masih mengandung kekurangan di sana-sini sehingga dengan segala kerendahan hati Penulis masih dan insya Allah akan tetap terus masih mengharapkan saran serta kritik yang membangun dari rekan-rekan pembaca.

Surabaya, Juli 2003

Penulis

UCAPAN TERIMA KASIH

Dengan tak bosannya mengucapkan syukur Alhamdulillah kepada Allah SWT, yang telah memberi terlalu banyak dari yang layak penulis terima, telah penulis curi sekelumit misteri-Mu, dan penulis kembalikan dalam bentuk buku ini. Di kesempatan ini, Penulis hendak menyampaikan rasa penghormatan yang setingginya serta rasa terima kasih kepada pihak-pihak yang telah memberi bantuan baik itu berupa moril maupun material dan langsung maupun tidak langsung kepada:

1. Terima kasih kepada nenek, bapak, ibu, adik, paman, bibi, dan keluarga yang telah memberi doa dan dukungan yang sangat besar kepada penulis selama ini hingga dapat menyelesaikan tugas akhirnya.
2. Bapak Dr. Ir. Arif Djunaidi M.Sc, selaku Dekan Fakultas Teknologi Informasi, semoga FTIF dapat terus maju dan berkembang.
3. Bapak Yudhi Purwananto S.Kom, M.Kom, selaku Ketua Jurusan sekaligus dosen pembimbing pertama yang telah memberi banyak masukan bagi penulis juga bimbingan dalam menyelesaikan tugas akhir. Semoga Teknik Informatika semakin maju dan berkembang di bawah kepemimpinan beliau.
4. Bapak Faizal Johan A S.Kom, selaku dosen pembimbing kedua yang telah memberikan ide bagi pengerjaan tugas akhir serta memberikan banyak bimbingan dan masukan bagi tugas akhir ini.
5. Bapak Ir. Khakim Ghazali, selaku dosen wali penulis yang telah membimbing penulis selama menjalani perkuliahan di jurusan ini.

6. Semua dosen jurusan Teknik Informatika yang telah membagikan ilmunya selama penulis menjalani masa perkuliahan.
7. Seluruh staf dan karyawan jurusan Teknik Informatika yang selalu siap membantu penulis dalam hal administrasi perkuliahan.
8. Aby Herwendo, dengan kesabarannya telah banyak membantu memberi masukan-masukan dalam penyelesaian tugas akhir ini.
9. Intan Yuniar Purbasari, yang selalu setia menemani dan menjadi inspirasi tersendiri bagi penulis dalam penyelesaian tugas akhir ini.
10. Rekan-rekan di PJB Pak Djati, Pak Sidik, Pak Bambang, Pak Wawan, Bu Ria, Mbak Yama, Mbak Dian, Mbak Siti, Mbak Oni, Mbak Dini, Mbak Dina, Mbak Maya, Mbak Diah, Mas Ilham, Mas Sobah, Mas Satrio, Mas Danang, Mas Andri, Mas Deni, Julius, Heri W, Arum, Agung, Andi, Zeri, yang selalu memberi nasehat dan dukungan kepada penulis, terima kasih atas semuanya.
11. Ratna, Ade, Citra, yang setia menemani penulis bersama-sama mengerjakan tugas akhir di laboratorium pemrograman.
12. Cphee, Didiet, Kendy, Joko, Caca, Deni, Yudha, Mbah Nur, Reza, Wong, Rossi, Iril, yang konsisten mengajak penulis bermain sepak bola tiap Selasa dan Jum'at untuk olahraga dan penghilang stres.
13. Teman-teman c0e lainnya yang selalu memberikan doa dan dukungannya kepada penulis. Tetap kompak selalu.
14. Rossi, Jakka, Joko dan teman-teman administrator labprog yang sudah banyak membantu penulis dengan menyediakan fasilitas laboratorium dalam menyelesaikan tugas akhirnya.

15. Teman-teman lain yang tidak dapat penulis sebutkan satu persatu dalam halaman ini.

Tiada untaian kata yang cukup yang dapat penulis sampaikan sebagai balas atas jasa yang penulis terima melainkan hanya harapan semoga ALLAH SWT membala semua amal tersebut. Jazakumullah Khairan Katsiran.

DAFTAR ISI

ABSTRAK	III
KATA PENGANTAR.....	IV
UCAPAN TERIMA KASIH.....	V
DAFTAR ISI.....	VIII
DAFTAR GAMBAR.....	XI
DAFTAR TABEL.....	XIII
BAB 1.....	1
PENDAHULUAN.....	1
1.1 <i>Latar Belakang</i>	1
1.2 <i>Tujuan</i>	2
1.3 <i>Perumusan Masalah.....</i>	3
1.4 <i>Batasan Masalah</i>	3
1.5 <i>Metodologi Pelaksanaan Tugas Akhir</i>	4
1.6 <i>Sistematika Penulisan</i>	5
BAB 2.....	7
FRAMEWORK .NET DAN REGULAR EXPRESSION.....	7
2.1 <i>Framework .NET</i>	7
2.2 <i>Windows Service.....</i>	10
2.3 <i>Web services.....</i>	11
2.3.1 Bagaimana <i>Web service</i> Bekerja	12
2.3.2 Membangun <i>Web services</i>	13
2.3.3 Utilitas <i>Web service</i>	14
2.3.4 <i>Service Description Language (SDL)</i>	14
2.3.5 <i>Discovery</i>	15
2.3.6 <i>Class Proxy</i>	16
2.4 <i>Multithreading.....</i>	16
2.5 <i>Regular expression</i>	18
2.5.1 <i>Regular expressions</i> dalam .NET Framework	18
2.5.2 <i>Regular expressions</i> sebagai Sebuah Bahasa	19
2.5.3 <i>Class-class Regular expression</i>	20
2.5.3.1 <i>Regex</i>	20
2.5.3.2 <i>Match</i>	20
2.5.3.3 <i>MatchCollection</i>	21
2.5.3.4 <i>GroupCollection</i>	22

2.5.3.5	CaptureCollection	23
2.5.3.6	Group.....	24
2.5.3.7	Capture	26
BAB 3.....		27
PERANCANGAN SISTEM		27
3.1	Arsitektur sistem	27
3.2	Use Case Diagram	28
3.2.1	Use Case Diagram <i>Administrator</i>	28
3.2.2	Use Case Diagram <i>Client</i>	29
3.3	Spesifikasi Pengguna.....	30
3.4	Activity Diagram	30
3.4.1	<i>Activity Diagram Pengambilan Berita</i>	31
3.4.2	<i>Activity Diagram Memanipulasi News Provider</i>	34
3.4.3	<i>Activity Diagram Memanipulasi Provider Directory</i>	35
3.4.4	<i>Activity Diagram Memanipulasi News Regex</i>	35
3.5	Perancangan Conceptual Diagram Model (CDM).....	37
3.6	Perancangan Physical Diagram Model (PDM)	38
3.7	Perancangan Class Diagram	39
3.8	Desain Database	40
3.8.1	Tabel NewsProvider	40
3.8.2	Tabel ProviderDirectory	40
3.8.3	Tabel NewsRegex	41
3.8.4	Tabel <i>LinksIndex</i>	42
3.8.5	<i>Relationship</i>	43
BAB 4.....		45
IMPLEMENTASI PERANGKAT LUNAK.....		45
4.1	Implementasi Rancangan Sistem.....	45
4.2	Perancangan Aplikasi <i>Administrator</i>.....	46
4.3	Perancangan Aplikasi <i>Client</i>	49
4.4	Pembuatan Windows Service	50
4.5	Perancangan Skenario <i>Regular expression</i>	53
4.5.1	Pengenalan situs	53
4.5.2	Pembuatan <i>regex</i> untuk <i>link</i> , <i>time</i> dan <i>title</i>	54
4.5.3	Pembuatan <i>regex</i> untuk <i>content</i>	54
BAB 5.....		55
UJI COBA DAN ANALISA HASIL.....		55
5.1	Lingkungan Uji Coba	55
5.1.1	<i>Server</i>	55
5.1.2	<i>Client</i>	56
5.2	Skenario Uji Coba	56
5.2.1	Uji Coba pada Situs Detik	57

5.2.1.1	Perancangan skenario <i>regex</i>	57
5.2.1.2	Pengambilan berita.....	58
5.2.2	Uji Coba pada Situs Republika	61
5.2.2.1	Perancangan skenario <i>regex</i>	61
5.2.2.2	Pengambilan berita.....	62
5.2.3	Uji Coba pada Situs Kompas	66
5.2.3.1	Perancangan skenario <i>regex</i>	67
5.2.3.2	Pengambilan berita.....	68
5.2.4	Uji Coba <i>Browsing</i>	73
BAB 6	77
KESIMPULAN DAN SARAN	77
6.1	Kesimpulan	77
6.2	Saran.....	78
DAFTAR PUSTAKA	79

DAFTAR GAMBAR

Gambar 2.1	<i>Framework .NET</i>	8
Gambar 2.2	Windows Service yang Sedang Berjalan di Komputer	11
Gambar 2.3	Arsitektur <i>Web service</i>	13
Gambar 2.4	SDL Contract dari <i>Web service</i>	15
Gambar 3.1	Gambar Arsitektur Sistem.....	28
Gambar 3.2	Use Case Diagram <i>Administrator</i>	29
Gambar 3.3	Use Case Diagram <i>Client</i>	29
Gambar 3.4	<i>Activity Diagram Pengambilan Berita</i>	34
Gambar 3.5	<i>Activity Diagram Memanipulasi News Provider</i>	35
Gambar 3.6	<i>Activity Diagram Memanipulasi Provider Directory</i>	36
Gambar 3.7	<i>Activity Diagram Memanipulasi News Regex</i>	36
Gambar 3.8	Diagram CDM.....	37
Gambar 3.9	Diagram PDM	38
Gambar 3.10	<i>Class Diagram</i>	39
Gambar 3.11	Tabel <i>NewsProvider</i>	40
Gambar 3.12	Tabel <i>ProviderDirectory</i>	41
Gambar 3.13	Tabel <i>NewsRegex</i>	42
Gambar 3.14	Tabel <i>LinksIndex</i>	43
Gambar 3.15	<i>Relationship</i>	44
Gambar 4.1	Tampilan Awal Aplikasi <i>News Administrator</i>	48
Gambar 4.2	Form <i>Directories</i>	48
Gambar 4.3	<i>Import Config</i>	48
Gambar 4.4	Tampilan Awal Aplikasi <i>Client</i>	49
Gambar 4.5	Form <i>Search</i> untuk Mencari Berita.....	50
Gambar 5.1	Data Direktori Detik.....	58
Gambar 5.2	Data <i>Regex</i> Detik.....	58
Gambar 5.3	Fungsi <i>ParseDirectory</i>	59
Gambar 5.4	Output Fungsi <i>parseDirectory</i> dalam format XML	60
Gambar 5.5	Tabel setelah <i>link-link</i> dan berita masuk	60
Gambar 5.6	Aplikasi <i>client-Melihat Isi Berita</i>	61
Gambar 5.7	Data Direktori Republika	62
Gambar 5.8	Data <i>Regex</i> Republika	63
Gambar 5.9	Fungsi <i>ParseDirectory</i>	63
Gambar 5.10	Output Fungsi <i>parseDirectory</i> dalam format XML	64
Gambar 5.11	Tabel setelah <i>link-link</i> dan berita masuk	65
Gambar 5.12	Aplikasi <i>client-Melihat Isi Berita</i>	66
Gambar 5.13	Data Direktori Kompas dan Kompas2	68
Gambar 5.14	Data <i>Regex</i> Kompas	68
Gambar 5.15	Data <i>RegEx</i> Kompas2	69
Gambar 5.16	Fungsi <i>ParseDirectory</i> pada Kompas	69

Gambar 5.17	Fungsi <i>ParseDirectory</i> pada Kompas2	69
Gambar 5.18	Output <i>parseDirectory</i> dalam XML untuk Kompas	70
Gambar 5.19	Output <i>parseDirectory</i> dalam XML untuk Kompas2	70
Gambar 5.20	Tabel setelah <i>link-link</i> dan berita masuk untuk Kompas	71
Gambar 5.21	Tabel setelah <i>link-link</i> dan berita masuk untuk Kompas2	72
Gambar 5.22	Aplikasi <i>client</i> -Melihat Isi Berita kompas	72
Gambar 5.23	Aplikasi <i>client</i> -Melihat Isi Berita kompas2	73
Gambar 5.24	Hasil Browsing Situs Detik	74
Gambar 5.25	ParseDirectory untuk Mengambil Detik <i>News</i>	75
Gambar 5.26	Link yang Bersangkutan Setelah Masuk ke Database	75
Gambar 5.27	Isi Link Berita Dibuka dengan Notepad.....	76

DAFTAR TABEL

Tabel 3.1 Proses Pengambilan Berita

31

BAB I

PENDAHULUAN

BAB 1

PENDAHULUAN

Dalam bab ini dijelaskan beberapa hal dasar yang meliputi latar belakang, permasalahan, batasan permasalahan, tujuan dan manfaat, metodologi pelaksanaan serta sistematika penulisan buku tugas akhir ini. Dari uraian tersebut diharapkan, gambaran umum permasalahan dan pemecahan yang diambil dapat dipahami dengan baik.

1.1 Latar Belakang

Belakangan ini, masyarakat banyak disuguhkan berbagai macam situs berita mulai dari yang *thin content* dengan kategori yang sedikit seperti *bintang-indonesia.com* sampai situs yang menjadi portal berita yang *rich content* masa kini dan sering diakses dengan bermacam-macam kategorinya semacam *detik.com*.

Berdasarkan pengamatan penulis, semakin lama situs *rich content* semakin memberikan ruang yang banyak untuk penempatan *banner* atau iklan dengan alasan untuk meningkatkan pendapatan situs. Sebagai akibatnya dalam mendapatkan informasi, pengguna harus terbebani oleh *banner* atau iklan tersebut. Sehingga kebanyakan pengguna hanya mengakses situs-situs berita tersebut untuk membaca berita terbaru yang umumnya tersaji melalui kepala berita (*headline*) berita pada halaman utama situs tersebut. Banyak diantara pengguna, terutama dari SOHO (*Small Office Home Office*) dengan *bandwidth* koneksi internet

terbatas, jarang mengakses berita utama dari *headline* yang sebagian besar digunakan untuk men-*download* gambar-gambar maupun *banner-banner* iklan dari situs berita yang kebanyakan belum tentu berguna dan tidak berhubungan dengan berita yang dibaca.

Sejalan dengan itu, proses pengambilan informasi secara keseluruhan dari situs berita secara otomatis ini akan memberikan kemudahan dan keefisienan bagi pengguna dalam memperoleh berita yang lengkap tanpa harus mengakses langsung situs-situs berita tersebut. Proses pengambilan informasi secara otomatis ini juga sangat memudahkan pengarsipan berita dan memahami kronologis kejadian dari berita tersebut.

1.2 Tujuan

Tujuan pembuatan tugas akhir ini adalah membuat perangkat lunak yang bekerja sebagai proses *background* (*background process*) yang melakukan pengambilan berita dari situs-situs berita yang telah ditentukan pengguna. Pengambilan dilakukan secara otomatis dan periodik tanpa harus mengakses langsung ke situs berita tersebut dengan melakukan penyaringan (*filtering*) terhadap situs berita sesuai skenario yang telah terdefinisi dalam *profile* yang dikenali melalui *regular expression*.

1.3 Perumusan Masalah

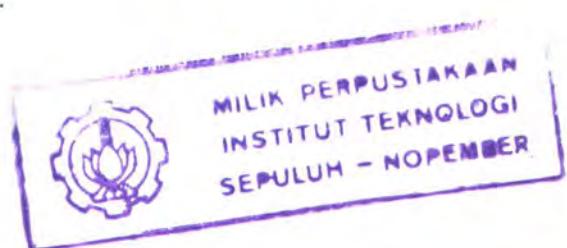
Permasalahan yang diangkat dalam tugas akhir ini adalah :

1. Bagaimana membuat suatu sistem aplikasi berbasis *web service* untuk men-*download* secara otomatis informasi berita utama dari satu atau lebih situs berita.
2. Bagaimana merancang dan membuat suatu skenario melalui *regular expression* yang mampu menyaring informasi utama dari situs berita yang memiliki berbagai macam jenis struktur baik *file* maupun datanya.
3. Bagaimana mempelajari dan mengetahui site map situs, metode pengambilan informasi, serta memberikan saringan (*filter*) yang sesuai pada situs-situs berita yang berbagai macam jenisnya tersebut.

1.4 Batasan Masalah

Dari permasalahan-permasalahan di atas, maka batasan dalam tugas akhir ini adalah:

1. Situs yang dijadikan bahan permasalahan adalah situs-situs yang memiliki Manajemen Isi Web (Web Content Management).
2. Aplikasi yang dihasilkan dari tugas akhir ini bukan aplikasi yang berbasis web.
3. Data yang dihasilkan berbasis text.
4. Pengambilan berita dilakukan secara statis, artinya jika struktur situs berubah maka skenario regesxnya pun harus diubah.
5. Pengambilan berita tidak termasuk gambar.



1.5 Metodologi Pelaksanaan Tugas Akhir

Pembuatan tugas akhir ini terbagi menjadi beberapa tahapan sebagai berikut:

1. Studi Literatur dan Basis Data yang ada

Tahap ini merupakan tahap pengumpulan informasi yang diperlukan untuk pembuatan perangkat lunak. Informasi yang dibutuhkan diperoleh dengan membaca literatur – literatur yang berhubungan dan mencari berbagai macam jenis situs berita yang memiliki *Content Management* seperti *detik.com*, *reuters.com*, dan sebagainya dan mempelajari pola manajemen isi situs tersebut maupun struktur penyimpanan file-filenya berdasarkan kategori-kategori yang ada pada situs tersebut.

Pada tahap ini diperlukan pemahaman masalah *Regular expression* dalam platform .NET khususnya pemrograman VB.NET melalui situs-situs yang tertera pada daftar pustaka. Juga mencari situs-situs yang membahas beberapa contoh *library regular expression* yang bisa digunakan dalam perancangan skenario nanti.

2. Pengembangan Aplikasi

Pada tahap ini dilakukan proses perancangan dan implementasi perangkat lunak. Selain itu, juga dilakukan proses pembuatan skenario melalui *Regular expression* untuk proses *filtering*.

3. Uji Coba dan Evaluasi

Pada tahap ini akan dilakukan evaluasi terhadap perangkat lunak yang telah selesai diimplementasikan untuk menemukan kesalahan – kesalahan yang mungkin terjadi serta melakukan perbaikan – perbaikan bagi

kesempurnaan perangkat lunak. Evaluasi juga dilakukan untuk melakukan pengambilan berita secara otomatis melalui situs-situs yang telah ditentukan.

4. Penyusunan Buku Tugas Akhir

Tahap penulisan buku Tugas Akhir bertujuan untuk menuliskan informasi – informasi mengenai perangkat lunak yang telah dibangun. Tahap ini sekaligus merupakan tahap terakhir dari penelitian yang dilakukan.

1.6 Sistematika Penulisan

Buku tugas akhir ini terdiri dari beberapa bab yang tersusun secara sistematis, yaitu :

BAB I PENDAHULUAN

Bab ini gambaran umum yang membahas latar belakang dan tujuan pembuatan perangkat lunak, serta permasalahan yang dihadapi dalam pengerjaan tugas akhir ini. Selain itu dijelaskan pula pembatasan terhadap masalah yang akan dihadapi serta metodologi yang dipakai untuk menyelesaiannya.

BAB II TEORI PENUNJANG

Bab ini membahas teori-teori dasar yang menunjang pengerjaan tugas akhir, meliputi pembahasan *framework .NET, regular expression, web service* dan *multithreading*, serta sedikit tentang VB.NET.

BAB III PERANCANGAN SISTEM

Dalam bab ini diuraikan mengenai sistem yang akan dibangun. Perancangan sistem ini dijelaskan dengan menggunakan notasi UML, yaitu mulai

pendefinisian *use case*, *activity diagram*, *component diagram* dan *class diagram*. Dalam bab ini juga dijelaskan perancangan *database* berbasis Microsoft Access.

BAB IV PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK

Dalam bab ini diuraikan perancangan dan pembuatan perangkat lunak yang terdiri dari :

- Pembuatan *Web service* berbasis *framework* Microsoft .NET
- Pembuatan aplikasi *Administrator* berbasis VB.NET
- Pembuatan aplikasi *Client* berbasis VB.NET
- Pembuatan *Windows service* untuk proses background berbasis VB.NET

BAB V UJI COBA DAN ANALISIS HASIL

Berisi tentang uji coba yang telah dilakukan terhadap perangkat lunak yang telah dibuat, beserta analisis dari hasil uji coba tersebut.

BAB VI PENUTUP

Berisi kesimpulan yang di dapat dari pembuatan tugas akhir ini dan dilengkapi dengan saran untuk kemungkinan pengembangan selanjutnya.

BAB II

FRAMEWORK.NET DAN REGULAR EXPRESSION

BAB 2

FRAMEWORK .NET DAN REGULAR EXPRESSION

Dalam bab ini dijelaskan mengenai metode dan *tool* yang dipakai dalam aplikasi ini, khususnya tentang *Framework .NET* dan *Regular expression*. Selain itu akan dijelaskan pula mengenai *web service* dan *multithreading*.

2.1 Framework .NET

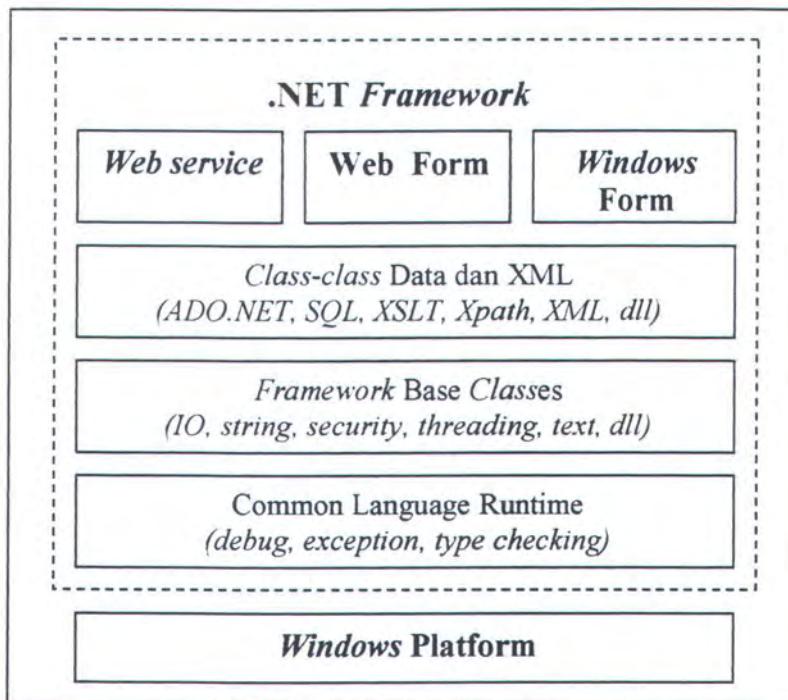
Framework .NET adalah sebuah *platform computing* baru yang dirancang untuk menyederhanakan pengembangan aplikasi pada lingkungan terdistribusi pada internet [08]. Bahasa yang didukung oleh *Framework .NET* adalah Visual Basic.NET, C++, ASP.NET, Jscript.NET dan C#.

Pada gambar 2.1, terlihat framework .NET berada diatas sistem operasi, yang berupa sistem operasi buatan Microsoft, yaitu *Windows 2000* (*Server* dan profesional), *Windows NT 4.0* (*Server* dan workstation), *Windows Millenium Edition*, *Windows 98* dan *Windows XP Profesional*

Framework .NET terdiri dari sejumlah komponen sebagai berikut :

a. Common Language Runtime (CLR)

CLR adalah komponen terpenting dalam *Framework .NET*. CLR menyediakan lingkungan eksekusi kode yang melakukan manajemen kode. Manajemen kode dapat berbentuk manajemen memori, manajemen *thread*, manajemen *security*, verifikasi kode dan kompilasi, dan layanan sistem lainnya.



Gambar 2.1 Framework .NET

CLR dapat digunakan oleh program .NET apapun, tanpa merujuk bahasa pemrograman tertentu.

Bahasa pemrograman .NET tidak melakukan kompilasi menjadi kode yang dieksekusi, melainkan melakukan kompilasi menjadi *intermediate code* yang disebut Microsoft Intermediate Language (MSIL). Kode MSIL kemudian dikirim ke CLR yang akan mengubah kode menjadi bahasa mesin yang dijalankan di mesin host.

b. *Framework Base Classes*

Framework Base Classes adalah *class library* yang dapat digunakan oleh setiap bahasa yang didukung oleh .NET. *Base classes* ini mengandung

sekumpulan *library* standar yang dibutuhkan, seperti *collection*, input/output, tipe data, dan *class* numerik. Sebagai tambahan, disediakan pula *class-class* untuk pengaksesan layanan-layanan sistem operasi, seperti grafik, jaringan, *thread*, dan akses data, juga sekumpulan *class* untuk pembuatan aplikasi berskala enterprise, seperti transaksi, *event*, dan *messaging*.

c. Class untuk Data dan XML

Mendukung manajemen data dan manipulasi XML. *Class-class* data mendukung manajemen data persisten, termasuk di dalamnya *class-class* SQL dan ADO.NET. *Framework .NET* juga menyediakan sejumlah *class* untuk memanipulasi data XML, melakukan pencarian serta penterjemahan XML.

d. Web service

Web service adalah entitas terprogram yang dapat diakses oleh banyak sistem terpisah melalui penggunaan standar internet, seperti XML dan HTTP. Sebuah *web service* dapat digunakan secara internal oleh sebuah aplikasi atau secara eksternal diekspos melalui internet untuk digunakan oleh banyak aplikasi.

Dengan kata lain, *web service* merupakan cara untuk mengakses fungsi-fungsi yang ada di *server* secara remote.

e. Web Forms

Menyediakan sejumlah *class* untuk pembuatan GUI (*Graphical User Interface*) untuk aplikasi web.

f. Windows Form

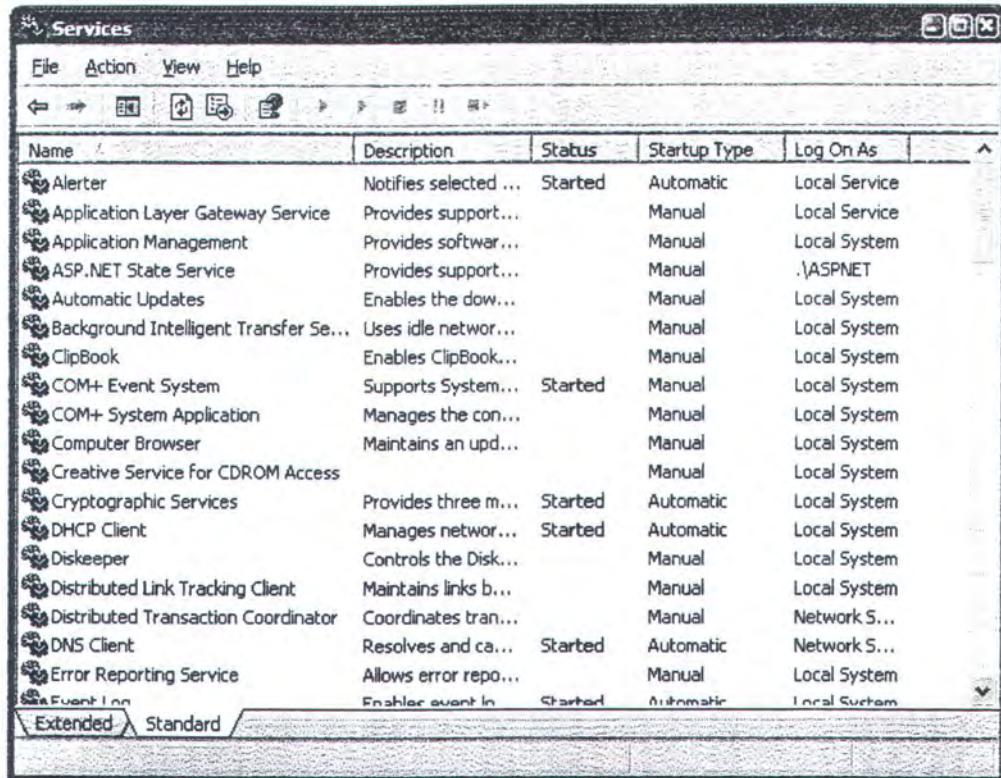
Menyediakan sejumlah *class* untuk pembuatan GUI untuk aplikasi *Windows*.

2.2 Windows Service

Windows service sebelumnya disebut *NT service*. Fungsi utama dari suatu *windows service* adalah untuk menjalankan aplikasi sebagai *background*. Ada beberapa hal yang membuat *windows service* berbeda dengan aplikasi windows biasa. *Windows service* mulai berjalan jauh sebelum pengguna melakukan *log/masuk* ke sistem (jika *windows service* tersebut diatur untuk mulai/*start* pada proses *boot-up*). Sebuah *windows service* juga dapat diatur sehingga pengguna harus melakukan *start* secara manual.

Windows service memiliki proses sendiri dan oleh karena itu ia dapat berjalan dengan sangat efisien. Secara normal, *windows service* tidak memiliki antarmuka untuk sebuah alasan sederhana yaitu memungkinkan *windows service* tetap berjalan walaupun tidak ada yang masuk/*log* ke dalam sistem, tapi ini bukanlah suatu keharusan, *windows service* juga dapat dibuat dengan antarmuka.

Dalam Windows 2000, daftar *service* yang saat ini sedang berjalan pada komputer dapat dilihat dengan membuka *Control Panel* → *Administrative Tools* → *Services*, seperti yang ditunjukkan pada gambar 2.2.



Gambar 2.2 Windows Service yang Sedang Berjalan di Komputer

2.3 Web services

Dengan semakin majunya teknologi, internet bukan lagi hanya digunakan untuk menampilkan halaman yang statis. Sekarang internet telah menjadi jembatan diantara banyak aplikasi, seperti *business-to-business* (B2B) *marketplace* dan *e-procurement*. Generasi baru dari internet ini akan merubah arsitektur dari aplikasi untuk menyediakan informasi kapan dan dimana saja kita membutuhkannya. Perusahaan-perusahaan website dapat mengekspos perangkat lunak mereka sebagai *service* di internet kepada partner bisnis mereka. *Service* inilah yang dinamakan *Web service* [09].

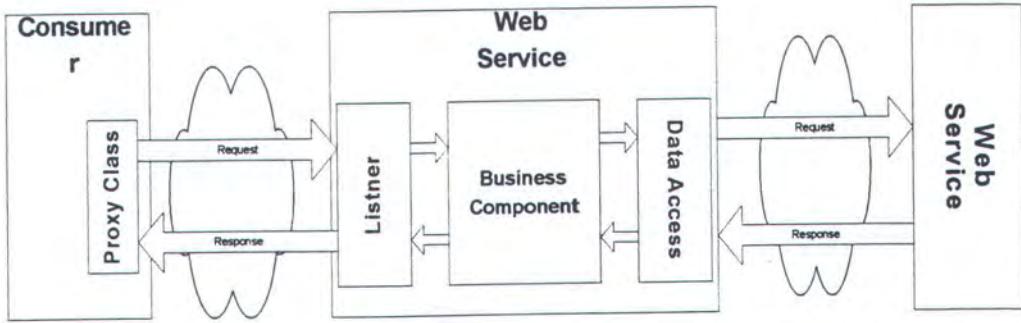
Suatu *Web service* adalah sebuah komponen yang menyediakan layanan untuk pelanggan yang menggunakan protokol internet standar (HTTP, XML) untuk mengakses layanan-layanan ini. *Web service ini* adalah suatu komponen bisnis yang bisa dikustomisasi dan tidak memiliki *user interface* dan dibuat hanya untuk dijalankan oleh program. Karena *web service* menggunakan HTTP, maka *web service* bersifat *firewall friendly* dan memiliki kelebihan seperti DCOM.

2.3.1 Bagaimana *Web service* Bekerja

Web service adalah komponen terprogram yang dapat diakses melalui protokol internet. *Web service* menggunakan XML dan *Simple Object Access Protocol* (SOAP) untuk berkomunikasi dengan pelanggan. XML menyediakan bahasa terstandarisasi untuk pertukaran data dalam format yang diterima secara luas. SOAP adalah protokol berbasis XML yang sederhana yang berjalan melalui HTTP untuk melakukan pertukaran data dalam lingkungan heterogen yang terdistribusi. Dengan kata lain SOAP = HTTP + XML.

Gambar 2.3 menunjukkan arsitektur dari *web service*. Konsumen mengirimkan request ke *web service* melalui internet menggunakan format pesan SOAP. Ketika pesan request SOAP datang, IIS menangkap pesan tersebut melalui TCP port 80, merutekan request kepada handler ASP.NET yang mengalokasikan *web service*, membuat komponen bisnis, memanggil *method* tertentu dalam objek, dan mengirimkan data. Komponen bisnis ini memproses request dan jika perlu mengambil data dari *database* atau dari *web service* yang lain. Komponen ini juga mengembalikan hasil ke ASP.NET yang kemudian diletakkan di dalam ‘kertas

surat' SOAP (*envelope*) dan mengirimkannya kembali ke consumer. Pada sisi consumer, .NET menyediakan sebuah *class proxy* yang mengkonversi pesan SOAP ini ke data. *Class proxy* ini juga meletakkan request ke dalam SOAP envelope dan mengirimkannya ke *web service*. SOAP merupakan protokol komunikasi yang default untuk *web service*. *Web service* juga bisa diakses dengan menggunakan protokol HTTP-GET dan HTTP-POST.



Gambar 2.3 Arsitektur *Web service*

Ketika menggunakan Visual Studio untuk membuat dan menggunakan *web service*, kita tidak perlu mengetahui arsitektur ini. .NET Framework mengkonversi semuanya, kita dapat membuat dan menggunakan *web service* tanpa terlebih dahulu harus mempelajari XML dan SOAP.

2.3.2 Membangun *Web services*

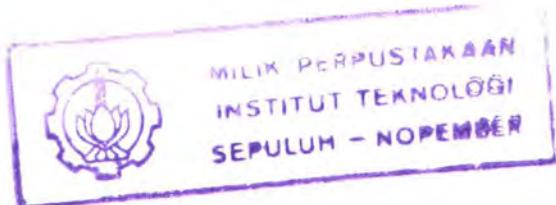
Karena *Web service* dapat diakses melalui Internet, maka disimpan dalam sebuah file dengan ekstensi .ASMX. Mirip dengan file .ASPX, file-file ini dikompilasi saat diakses untuk pertama kalinya.

2.3.3 Utilitas *Web service*

Pengguna *Web service* dapat mengetahui mengenai *Web service* tersebut dengan mengetahui metode yang dipakai, parameter yang diperlukan, dan protokol yang digunakan. Pengguna bisa menggunakan komponen ini tanpa perlu mengetahui arsitekturnya karena .NET *Framework* telah menyediakannya. ASP.NET menjelaskan semua metode dan parameter dalam *Web service* melalui *Service Description Language* (SDL). Bahkan, ASP.NET memungkinkan untuk ditemukannya semua *Web service* yang tersedia di *Web server*. ASP.NET juga membuat sebuah *class proxy*. Dibawah ini akan dijelaskan bagaimana ASP.NET melakukan hal tersebut.

2.3.4 *Service Description Language* (SDL)

Sebuah *Web service* bisa dimintai daftar metode dan meresponnya dengan deskripsi dalam format yang mudah dimengerti. SDL menjelaskan format pesan yang dimengerti oleh *Web service*. SDL *contract* menggunakan format XML untuk menjelaskan protokol yang digunakan oleh *Web service* (SOAP, HTTP-GET, HTTP-POST), metode yang tidak terlihat, dan masukan (request) dan keluaran (response) dari metode tersebut. SDL mirip dengan *type library* dalam sebuah obyek COM. Untuk meminta *Web service* mengembalikan SDL *contract*, digunakan *query string*?SDL pada URL *Web service*. Dalam contoh ini, URL-nya adalah <http://localhost/Chapter10/MyWebService.asmx?WSDL>. Gambar 2.4 menunjukkan SDL *contract* dari *Web service*. Alternatif lainnya, SDL *contract* bisa dilihat dengan menekan link saat mengakses *Web service* pada sebuah browser.



The screenshot shows a Microsoft Internet Explorer window with the title bar "http://localhost/Chapter10/mywebservice.asmx?WSDL - Microsoft Internet Explorer". The address bar also displays the same URL. The main content area contains the XML code of the WSDL (Web Services Description Language) document. The XML defines service operations like HelloWorld and GetOrders, their responses, and the data types used.

```

<?xml version="1.0" encoding="utf-8" ?>
- <definitions xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:soapenc="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  targetNamespace="http://tempuri.org/" xmlns:a="http://schemas.xmlsoap.org/wsdl/">
- <types>
  - <s:schema attributeFormDefault="qualified" elementFormDefault="qualified"
    targetNamespace="http://tempuri.org/">
    <s:import namespace="http://www.w3.org/2001/XMLSchema" />
  - <s:element name="HelloWorld">
    <s:complexType />
  </s:element>
  - <s:element name="HelloWorldResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="HelloWorldResult" nillable="true"
          type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
  - <s:element name="GetOrders">
    <s:complexType>
      <s:sequence>
    </s:sequence>
  </s:element>
- <s:element name="GetOrderDetails">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="GetOrderDetailsResult" nillable="true"
        type="s:string" />
    </s:sequence>
  </s:complexType>

```

Gambar 2.4 SDL Contract dari *Web service*

2.3.5 Discovery

SDL berguna jika diketahui *Web service* mana yang diinginkan. *Web service* *Discovery* membantu meletakkan dan menanyakan deskripsi *Web service*. Setiap Web site mempublikasikan semua *Web service* miliknya dalam sebuah file .VSDISCO. File ini adalah sebuah dokumen XML yang berisi URL-URL dari semua deskripsi SDL. Dengan proses *discovery* ini, konsumen mengetahui keberadaan sebuah *Web service*, kemampuan yang dimiliki dan bagaimana berinteraksi dengannya.

2.3.6 Class Proxy

Pengguna Web harus mengirim pesan ke *Web service* menggunakan SOAP. SOAP *marshalling* dapat dibuat untuk mengirim dan menerima data dari *Web service* melalui HTTP, atau bisa juga digunakan .NET *Framework* untuk membuat sebuah *class proxy* yang berisi *network invocation* yang sesuai dan kode *marshalling* untuk mendapatkan (*invoke*) dan menerima respon dari *Web service*. *class proxy* ini bisa direferensikan dalam program *client* dan digunakan untuk mendapatkan *Web service* sama seperti mendapatkan metode lokal.

Jika digunakan Visual Studio untuk membuat dan menggunakan *Web service*, .NET *framework* telah melakukan hal tersebut diatas.

2.4 Multithreading

Multithreading memungkinkan suatu aplikasi memproses lebih dari satu pekerjaan pada saat yang bersamaan [07]. Saat menggunakan *multithreading*, satu *thread* memproses *user interface* 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.

2.5 Regular expression

Regular expression menyediakan metode yang kuat, fleksibel dan efisien untuk memproses teks [04]. 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.

2.5.1 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

2.5.2 *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.

2.5.3 *Class-class Regular expression*

Bagian ini menjelaskan kelas-kelas *regular expression* .NET Framework [04]:

2.5.3.1 *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.

2.5.3.2 *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.

```
// Instantiate sebuah obyek Regex.  
Regex r = new Regex("abc");
```

```
// Mencari kecocokan dalam string.
Match m = r.Match("123abc456");
if (m.Success)
{
    // Mencetak posisi karakter dimana ditemukan sebuah kecocokan.
    // (Posisi karakter 3 dalam contoh ini).
    Console.WriteLine("Found match at position " + m.Index);
}
```

2.5.3.3 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];

// Instantiate sebuah obyek Regex baru dan menentukan regular expression.
Regex r = new Regex("abc");
// Menggunakan metode Matches untuk menemukan semua kecocokan dalam input string.
mc = r.Matches("123abc4abcd");
// Ulangi melalui kumpulan kecocokan untuk mengambil semuanya
// kecocokan dan posisi
for (int i = 0; i < mc.Count; i++)
{
    // Menambahkan string yang cocok ke string array.
    results[i] = mc[i].Value;
    // Menyimpan posisi karakter dimana ditemukan kecocokan.
    matchposition[i] = mc[i].Index;
}
```

2.5.3.4 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

2.5.3.5 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 *console* 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;

        // Mencari pengelompokan dari "Abc".
    }
}
```

```

Regex r = new Regex("(Abc)+");
// Menentukan string yang dicari.
m = r.Match("XYZAbcAbcAbcXYZAbcAb");
gc = m.Groups;

// Mencetak jumlah grup.
Console.WriteLine("Captured groups = " + gc.Count.ToString());

// Ulangi masing-masing grup.
for (int i=0; i < gc.Count; i++)
{
    cc = gc[i].Captures;
    counter = cc.Count;

    // Mencetak jumlah captures dalam grup ini.
    Console.WriteLine("Captures count = " + counter.ToString());

    // Ulangi masing-masing capture dalam grup.
    for (int ii = 0; ii < counter; ii++)
    {
        // Mencetak capture dan posisi.
        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

```

2.5.3.6 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];
// Perentuan substrings abc, ab, b.
Regex r = new Regex("(a(b))c");
Match m = r.Match("abdabc");
for (int i = 0; m.Groups[i].Value != ""; i++)
{
    // Menyalin grup ke dalam string array.
    results[i]=m.Groups[i].Value;
    // Menyimpan posisi karakter.
    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+)");
```



MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

```
Match m = r.Match("Section1:119900");
```

Regular expression ini menghasilkan output sebagai berikut.

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

2.5.3.7 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("bcababcabc")  
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

PERANCANGAN SISTEM

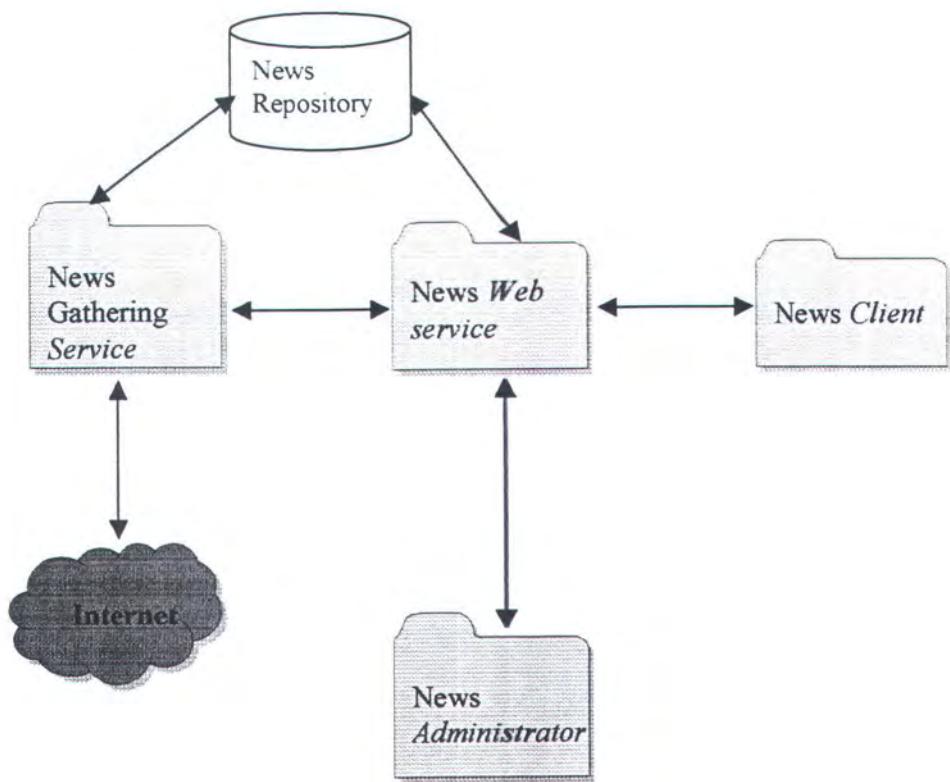
BAB 3

PERANCANGAN SISTEM

Pada bab ini akan dibahas mengenai perancangan sistem mulai dari desain *database* sampai desain diagram-diagram UML-nya. Sebelum menginjak ke bagian pertama, perlu diketahui bahwa perangkat lunak ini sendiri terdiri dari empat bagian, yaitu *News Administrator*, *News Web service*, *News Client* dan *News Gathering Service*.

3.1 Arsitektur sistem

News Administrator mengatur segala hal yang berhubungan dengan *inserting*, *updating* dan *deleting* data ke *database*. Aplikasi inilah yang menangani data-data baru yang akan masuk ke sistem, seperti situs baru, alamat URL dan sebagainya. Demikian juga jika ada perubahan maupun penghapusan terhadap *database*. *News Web service* berfungsi sebagai *service* yang mengantarkan *server* dan *client*. Fungsi-fungsi yang digunakan *client* umumnya yang berhubungan dengan akses ke *database*, sedangkan *server* menggunakan fungsi-fungsi untuk *parsing (filtering)*. *News Client* merupakan aplikasi bagi pengguna (*client*) berisi antarmuka untuk membaca dan mencari berita. *News Gathering Service* berfungsi sebagai *server* yang bertugas mengambil berita secara berkala. Aplikasi ini merupakan *windows service* yang berjalan sebagai proses *background* pada saat mengambil berita. Gambar 3.1 menunjukkan arsitektur sistem:



Gambar 3.1 Gambar Arsitektur Sistem

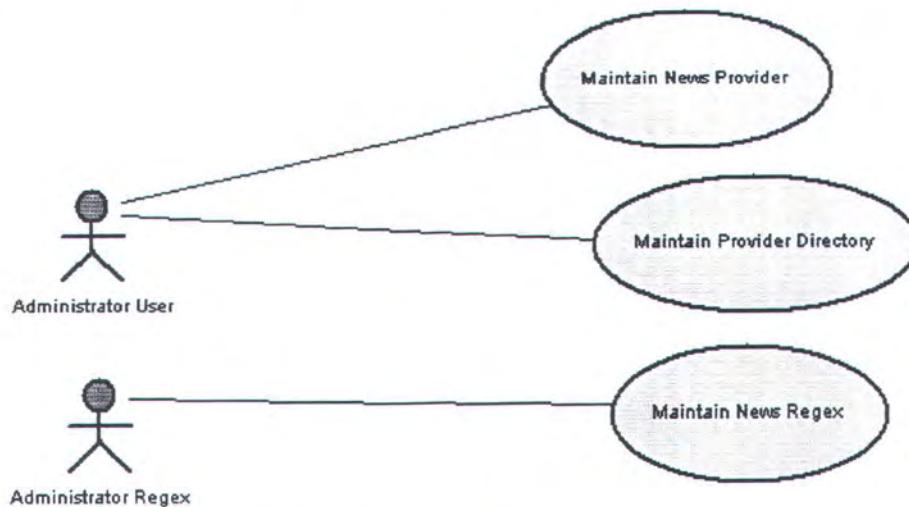
3.2 Use Case Diagram

Use case diagram ini akan menggambarkan hubungan aktor dengan proses-proses yang menjadi tanggung jawabnya. Dalam use case diagram ini ada dua aktor, yaitu *Administrator* dan *Client*. Masing-masing aktor memiliki tanggung jawab masing-masing seperti yang dijelaskan sebagai berikut:

3.2.1 Use Case Diagram *Administrator*

Aktor *Administrator* di sini dibagi menjadi dua, yaitu *Administrator User* dan *Administrator Regex*. *Administrator User* bertanggung jawab mengelola data-data

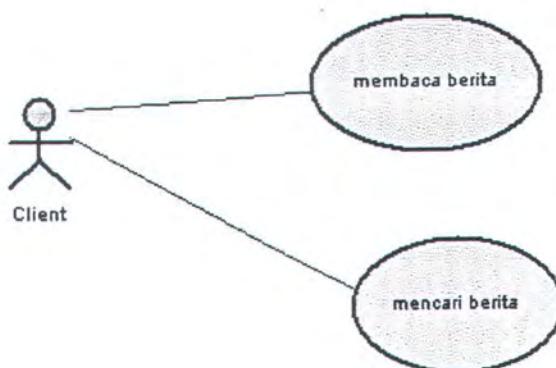
yang berhubungan dengan tabel-tabel *NewsProvider* dan *ProviderDirectory*. Sedangkan *Administrator Regex* bertanggung jawab untuk membuat, menambah, mengubah atau menghapus formula *regex*.



Gambar 3.2 Use Case Diagram *Administrator*

3.2.2 Use Case Diagram *Client*

Use case diagram *Client* yang ditunjukkan pada gambar 3.3 ini menggambarkan proses-proses yang dilakukan oleh aktor *Client*, diantaranya adalah membaca dan mencari berita.



Gambar 3.3 Use Case Diagram *Client*

3.3 Spesifikasi Pengguna

Dalam penjelasan diagram *use case* di atas ada tiga macam pengguna, yaitu *Administrator User*, *Administrator Regex* dan *Client*. Spesifikasi atau *skill* yang diperlukan untuk ketiga pengguna tersebut masing-masing adalah:

- ❖ *Administrator User* : Pengguna ini berfungsi sebagai operator yang bertugas memasukkan data-data ke dalam database, maka *skill* yang diperlukan adalah *entry data*.
- ❖ *Administrator Regex* : Pengguna ini bertugas menganalisis situs berita dan membuat formula *regex*-nya. *Skill* yang diperlukan adalah mengerti *Regular Expression* dan *HTML*.
- ❖ *Client* : Pengguna ini adalah pengguna yang mengakses isi berita. Pengguna ini berfungsi sebagai *client* yang hanya memerlukan informasi utama dari situs berita tanpa mengutamakan iklan/*banner*, gambar, dan menu/*link* lainnya, hanya berita utamanya saja.

3.4 Activity Diagram

Activity diagram menunjukkan aliran proses yang terjadi pada perangkat lunak ini. Ada beberapa *activity* diagram yang akan dijelaskan sebagai berikut:

3.4.1 Activity Diagram Pengambilan Berita

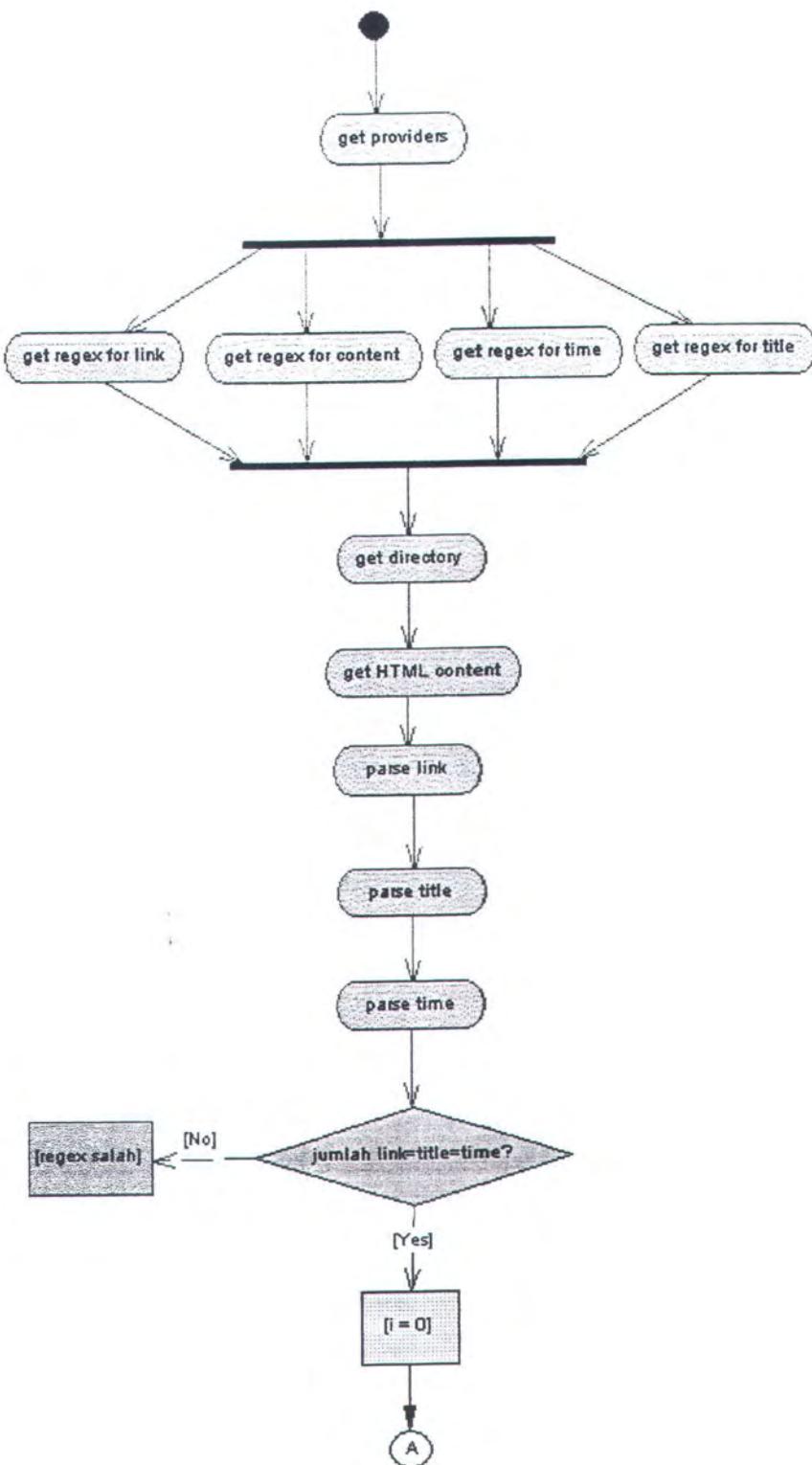
Diagram ini menunjukkan aliran proses utama yang terjadi pada perangkat lunak ini, yaitu proses pengambilan berita. Proses ini dilakukan sebagai proses background (*windows service*) pada *server*. Gambar 3.4 menunjukkan aliran proses yang terjadi saat *service* melakukan pengambilan berita.

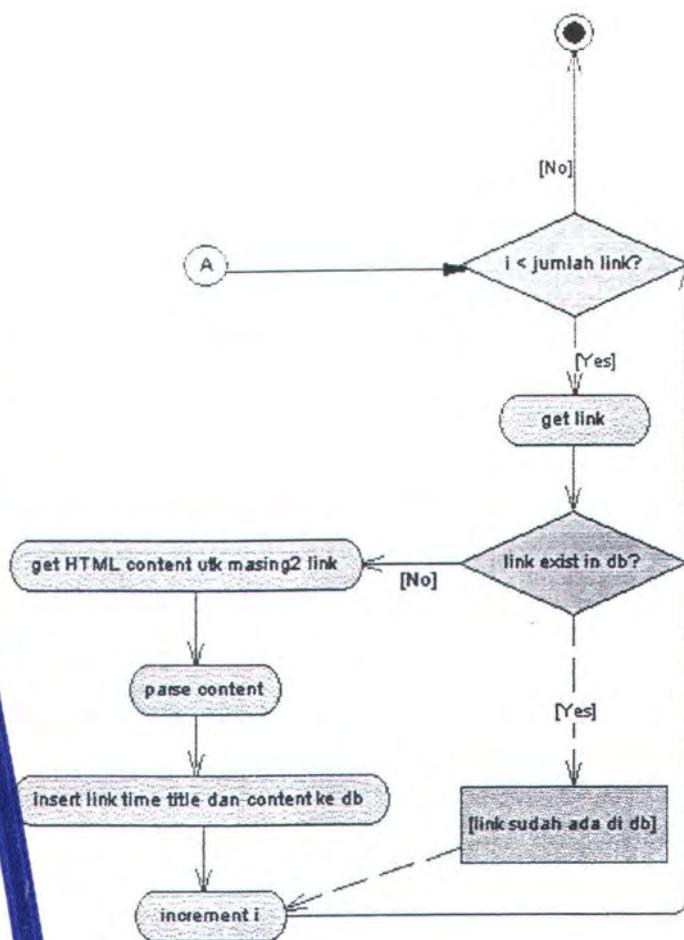
Tabel 3.1 di bawah ini akan dijelaskan masing-masing proses yang terjadi dalam *activity diagram* ini:

Tabel 3.1 Proses Pengambilan Berita

Nama Proses	Deskripsi
Get providers	Untuk mendapatkan <i>array</i> dari <i>class NewsProvider</i> yang berisi atribut-atribut dari tabel <i>NewsProvider</i> beserta direktori-direktori dari tabel <i>ProviderDirectory</i> dan formula-formula <i>regex</i> -nya dari tabel <i>NewsRegex</i>
Get <i>regex</i> for <i>link</i> , <i>content</i> , <i>time</i> , dan <i>title</i>	Untuk mendapatkan satu <i>record</i> yang berisi <i>class newsRegex</i> diambil dari tabel <i>newsRegex</i> beserta atribut-atributnya. <i>Regex</i> yang diambil sesuai nama proses, misalnya <i>get regex for link</i> berarti mengambil <i>regex</i> untuk <i>link</i>
Get directory	Untuk mendapatkan satu <i>record</i> yang berisi <i>class providerDirectory</i> diambil dari tabel <i>providerDirectory</i> beserta atribut-atributnya
Get HTML content	Untuk mengambil <i>HTML source</i> dari suatu <i>URL</i> . Dalam hal ini mengambil URL dari suatu direktori dari penyedia berita tertentu sesuai urutan <i>looping</i> -nya
Parse <i>link</i> , <i>title</i> , dan <i>time</i>	Dari <i>regex</i> dan <i>HTML source</i> tadi dilakukan <i>parsing/filtering</i> dengan melakukan

	<i>matching/replacing</i> menggunakan <i>regex</i> tersebut sehingga menghasilkan hasil yang diinginkan. Parse link untuk parsing link, dan seterusnya. Hasilnya disimpan dalam bentuk <i>array</i>
Get link	Untuk mendapatkan data-data <i>link</i> hasil <i>parsing</i> dalam bentuk <i>array</i> tersebut yaitu nama penyedia berita, direktori, judul berita, <i>link's URL</i> , waktu berita, dan waktu masuk
Get HTML content utk masing2 link	Dari data-data tersebut diambil <i>link's URL</i> -nya kemudian di ambil <i>HTML source</i> -nya
Parse content	Dari <i>HTML source</i> tersebut dilakukan <i>parsing</i> untuk mengambil <i>content</i> -nya
Insert link time title dan content ke db	Setelah semuanya selesai, maka data-data tersebut dimasukkan ke <i>database</i>
Increment i	Naikkan nilai "i" untuk <i>looping</i>

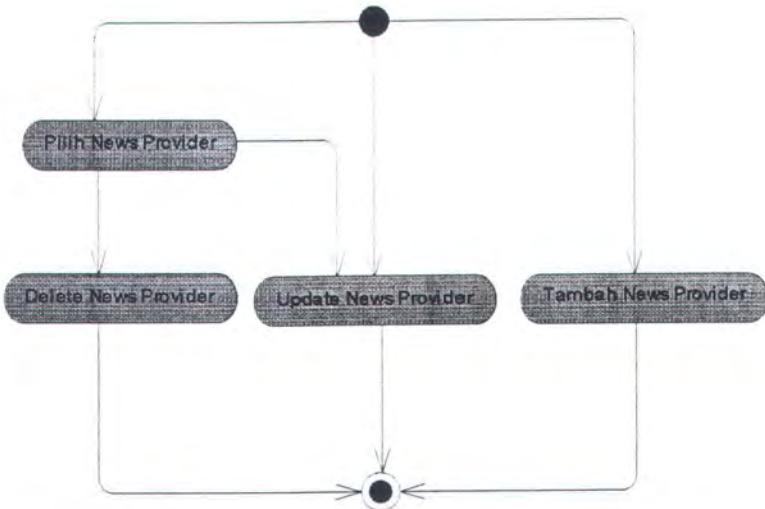




Gambar 3.4 Activity Diagram Pengambilan Berita

3.4.2 Activity Diagram Memanipulasi *News Provider*

Diagram ini menunjukkan proses manipulasi tabel *NewsProvider* yang meliputi proses *inserting*, *updating* dan *deleting record-record* yang berhubungan dengan data-data penyedia berita (*News Provider*). Proses ini terjadi pada aplikasi *Administrator*. Gambar 3.5 di bawah menjelaskan proses tersebut.



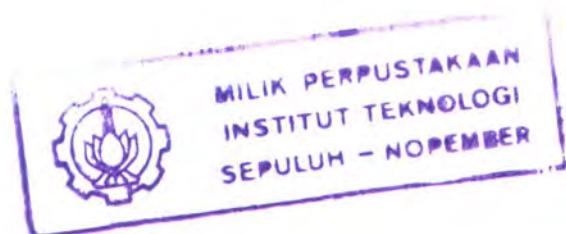
Gambar 3.5 Activity Diagram Memanipulasi News Provider

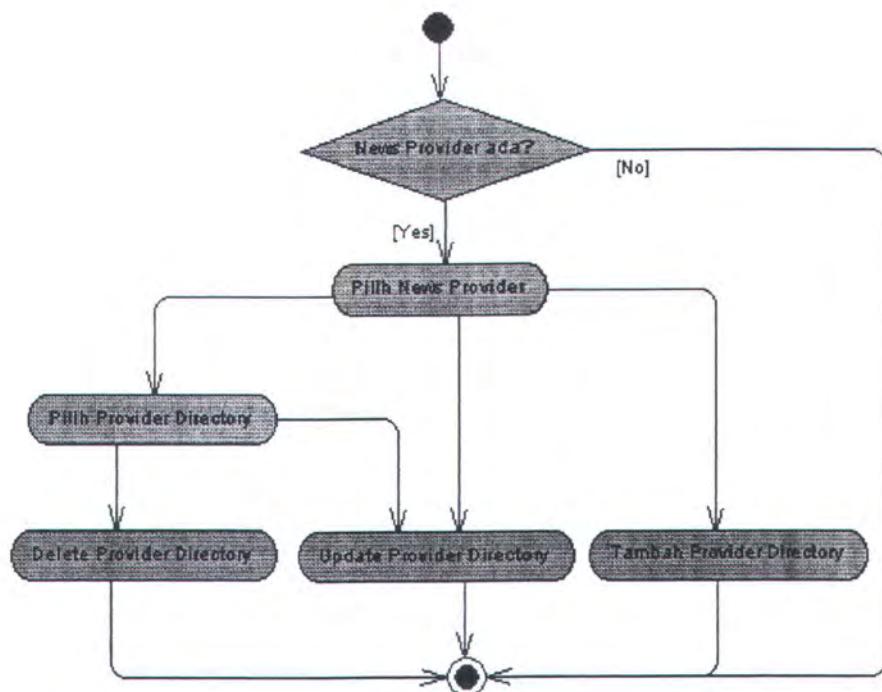
3.4.3 Activity Diagram Memanipulasi *Provider Directory*

Diagram ini menunjukkan proses manipulasi tabel *ProviderDirectory* yang meliputi proses *inserting*, *updating* dan *deleting record-record* yang berhubungan dengan data-data direktori tertentu dari satu penyedia berita (*Provider Directory*). Proses ini juga terjadi pada aplikasi *Administrator*. Gambar 3.6 menjelaskan proses tersebut.

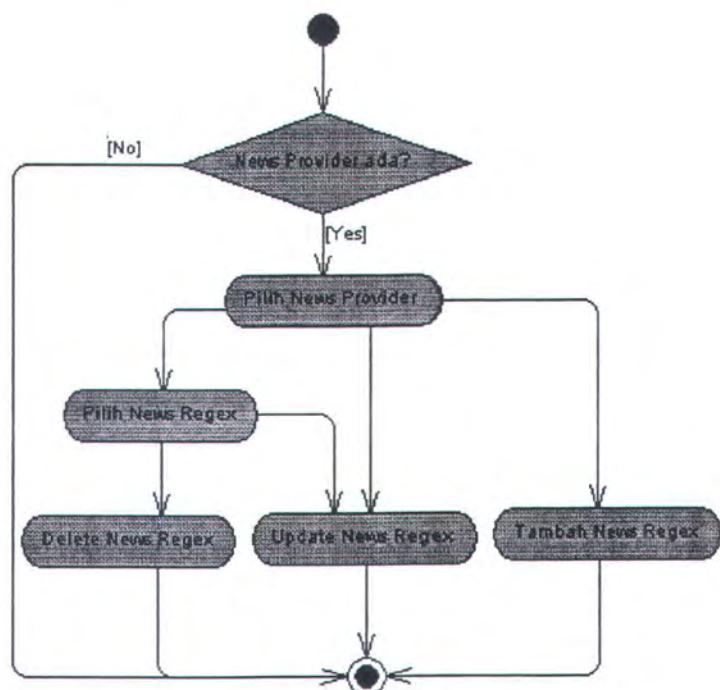
3.4.4 Activity Diagram Memanipulasi *News Regex*

Diagram ini menunjukkan proses manipulasi tabel *NewsRegex* yang meliputi proses *inserting*, *updating* dan *deleting record-record* yang berhubungan dengan data-data formula *regex* dari satu direktori tertentu dari satu penyedia berita (*News Regex*). Proses ini juga terjadi pada aplikasi *Administrator*. Gambar 3.7 menjelaskan proses tersebut.





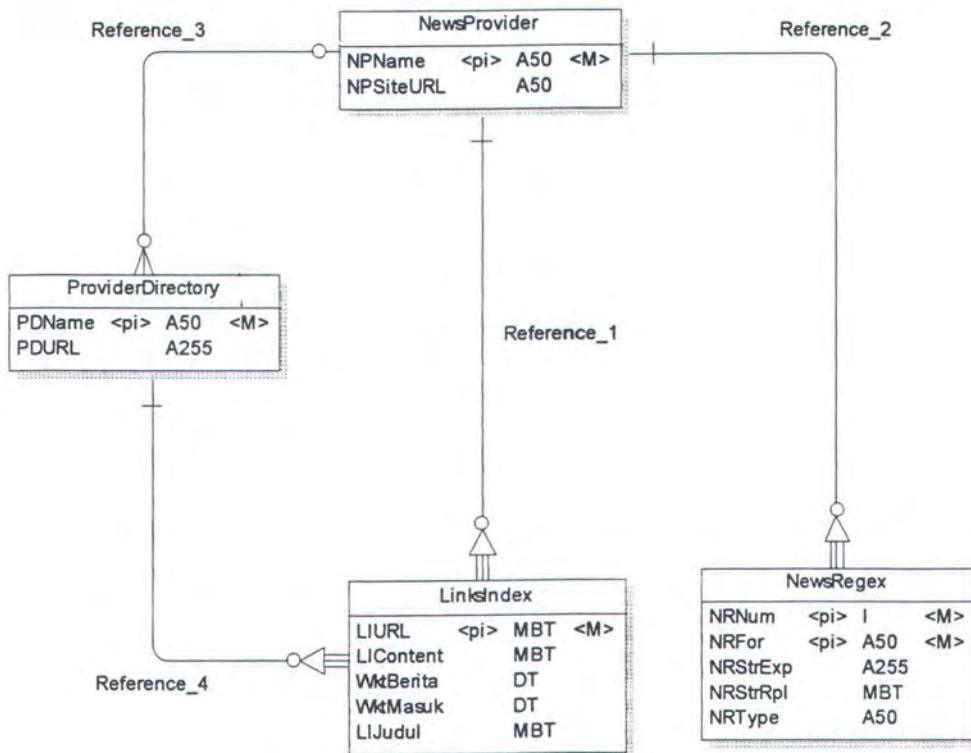
Gambar 3.6 Activity Diagram Memanipulasi Provider Directory



Gambar 3.7 Activity Diagram Memanipulasi News Regex

3.5 Perancangan Conceptual Diagram Model (CDM)

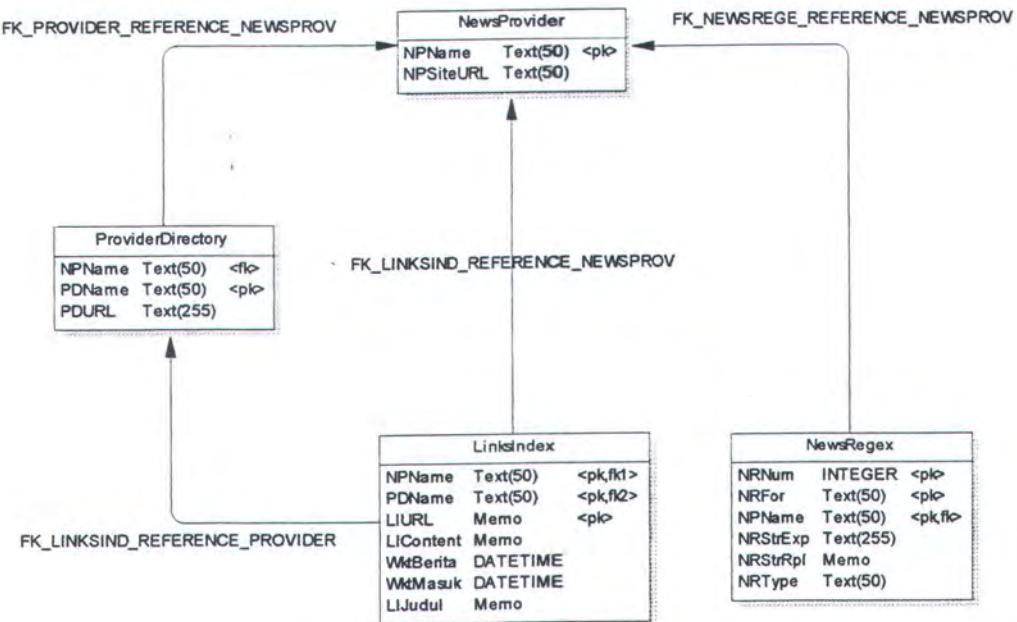
Rancangan *conceptual diagram* (CDM) dari *database* sistem perangkat lunak ini, dapat dilihat pada gambar 3.8. CDM ini menunjukkan hubungan antar entitas dan atribut-atributnya. Seperti dapat dilihat pada gambar tersebut bahwa entitas *NewsProvider* merupakan entitas utama dalam model tersebut. Karena entitas ini menjadi *primary key* bagi entitas-entitas lain. Entitas *NewsProvider* ini akan menentukan proses tahap selanjutnya seperti pencarian direktori (atribut PDName) dari entitas *ProviderDirectory*, pengambilan *link-link* dari entitas *LinksIndex* dan sebagainya.



Gambar 3.8 Diagram CDM

3.6 Perancangan Physical Diagram Model (PDM)

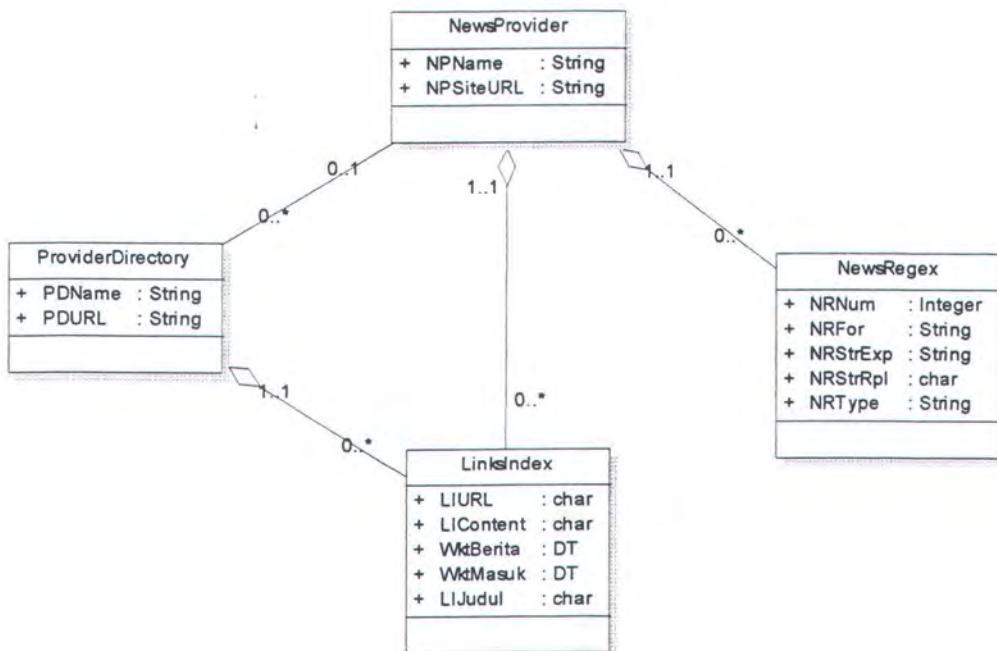
Diagram PDM hasil *generate* dari diagram CDM ditunjukkan pada gambar 3.9. Dalam diagram ini terdapat hubungan/relasi antar entitas. Seperti hubungan antara entitas *NewsProvider* dan *ProviderDirectory* yaitu *one-to-many* dengan *half-partisipation*. *One-to-many* di sini berarti satu *NewsProvider* (penyedia berita) bisa memiliki banyak (satu/beberapa) *ProviderDirectory*. *Half-partisipation* di sini berarti *NewsProvider* bisa tidak memiliki *ProviderDirectory*. Sedangkan *full-partisipation* seperti pada hubungan antara entitas *NewsProvider* dengan *LinksIndex* berarti tiap *LinksIndex* harus memiliki *NewsProvider*, sedangkan sebaliknya, *NewsProvider* tidak harus memiliki *LinksIndex* karena *half-partisipation*.



Gambar 3.9 Diagram PDM

3.7 Perancangan Class Diagram

Rancangan *class diagram* pada gambar 3.10 di bawah ini lebih menunjukkan hubungan (*relationship*) antar tabel dalam *database*. Diagram ini juga hasil generate diagram PDM ke *class diagram* melalui tools dari Power Designer 9. seperti pada PDM sebelumnya, diagram ini juga memiliki hubungan/relasi antar entitas, hanya simbolnya saja yang berbeda. Seperti simbol *0..1* berarti *NewsProvider* yang berpartisipasi minimum 0 dan maksimum 1. Sedang simbol *0..n* berarti minimum 0 dan maksimum tak terhingga. Simbol *diamond* menunjukkan hubungan *dependency*, artinya tabel yang ber-*dependency* akan berpengaruh apabila entitas *master*-nya mengalami perubahan, yaitu *inserting*, *updating*, dan *deleting*. Seperti relasi antara *NewsProvider* dan *NewRegex*, entitas *NewsProvider* adalah entitas *master* bagi *NewRegex*.



Gambar 3.10 Class Diagram

3.8 Desain Database

Bagian ini merupakan penjelasan dari tabel-tabel dalam *database* beserta atribut-atributnya. Tabel-tabel tersebut diantaranya adalah sebagai berikut:

3.8.1 Tabel NewsProvider

Tabel ini berisi data-data tentang penyedia berita. Atribut-atributnya adalah *NPName* yang menunjukkan nama situs berita, sedangkan *NPSiteURL* menunjukkan alamat URL dari situs tersebut.

NewsProvider : Table		
	NPName	NPSiteURL
+ ade	situs	
► + Detik		http://www.detik.com/index.html
+ JawaPos		http://www.jawapos.com
+ kamil		http://www.kamil.com
+ Kompas		http://www.kompas.co.id/utama/index.cfm
+ nama	situs	
+ Surya		http://www.surya.co.id/29032003/list.phtml

Gambar 3.11 Tabel NewsProvider

3.8.2 Tabel ProviderDirectory

Tabel ini berisi data-data tentang direktori dari penyedia berita. Atribut-atributnya adalah :

NPName : nama situs berita

PDName : nama direktori

PDURL : alamat URL dari direktori

ProviderDirectory : Table			
	NPName	PDName	PDURL
► + Detik	Finance		http://www1.detik.com/indexberita/index.cfm?fuseaction=index.detikFinance
+ Detik	food		http://www1.detik.com/indexberita/index.cfm?fuseaction=index.detikFood
+ Detik	Hot		http://www1.detik.com/indexberita/index.cfm?fuseaction=index.detikHot
+ Detik	i-Net		http://www1.detik.com/indexberita/index.cfm?fuseaction=index.detiki-Net
+ Detik	news		http://www1.detik.com/indexberita/index.cfm?fuseaction=index.detikNews
+ Detik	Sports		http://www1.detik.com/indexberita/index.cfm?fuseaction=index.detikSports
+ Kompas	Ekonomi		http://www.kompas.co.id/bisnis/bisnis.cfm???
+ Kompas	Gaya Hidup		http://www.kompas.co.id/gayahidup/index.htm???
+ Kompas	Indeks		http://www.kompas.co.id/utama/index.cfm???
+ Kompas	Kesehatan		http://www.kompas.co.id/kesehatan/index.htm???
+ Kompas	Metropolitan		http://www.kompas.co.id/metro/index.htm???
+ Kompas	Olah Raga		http://www.kompas.co.id/olahraga/index.htm???
+ Kompas	Teknologi		http://www.kompas.co.id/teknologi/index.htm???
+ Surya	cerpen		http://www.surya.co.id/16032003/cerpen.shtml
+ Surya	daerah		http://www.surya.co.id/29032003/daerah.shtml
+ Surya	dokumentasi surya		http://202.154.62.70/
+ Surya	ekonomi-bisnis		http://www.surya.co.id/29032003/bisnis.shtml
+ Surya	galeri foto		http://www.surya.co.id/29032003/galerifoto.shtml
+ Surya	hiburan		http://www.surya.co.id/29032003/hiburan.shtml
+ Surya	horoskop		http://www.surya.co.id/16032003/horoskop.shtml
+ Surya	interupsi cak		http://localhost/surya/22052003/interupsi.shtml

Gambar 3.12 Tabel ProviderDirectory

3.8.3 Tabel NewsRegex

Tabel ini berisi data-data tentang formula *regex* dari penyedia berita. Atribut-atributnya adalah :

NPNum : nomor *regex*, menunjukkan urutan formula tersebut dieksekusi oleh perangkat lunak

NPFor : fungsi *regex*, menunjukkan fungsi *regex* tersebut, yaitu untuk mem-filter *time*, *title*, *link* atau *content*.

NPName : nama penyedia berita

NRStrExp : formula *regex* (*regex's expression*)

NRStrRpl : karakter pengganti (*replacement character*) untuk tipe *regex* tipe *replace* (*regex.replace*)

NRTyp : tipe *regex* (*match* atau *replace*)

NewsRegex : Table						
NRNum	NRFor	NPName	NRStrExp	NRStrRpl	NRTtype	
1	1	Detik	(^.*<BODY>		1	
	1 2	Detik			1	
	1 3	Detik	tanggalindex.*?>[0-9]*(< ")		2	
	1 4	Detik	</?b>		1	
	2 1	Detik	/blockquote.*		1	
	2 2	Detik	(\s\s+) \t		1	
	2 4	Detik	</?i>		1	
	3 1	Detik	(^")blockquote		1	
	3 2	Detik	<"/?([bB][Rr])<"/>		1	
	3 4	Detik	(? font class="judulindex.*?>(.?")<		2	
	4 1	Detik	<A[^>]*?HREF<=\\s*[\"?](\\^>]+)[\"?]<		2	
	4 2	Detik	^texberita.*?>		1	
	5 2	Detik	<font\\s*> *		1	
	6 2	Detik	<[Pp]>	□□	1	
	7 2	Detik	</>^>		1	
	1 1	Kompas	(? <!--START UPDATE-->)(.*?)(?:Index Berita ya		2	
	1 2	Kompas	<!--Start-->(.?")<!--fotonews-->		2	
	1 3	Kompas	(? <!--START UPDATE-->)(.*?)(?:Index Berita ya		2	
	1 4	Kompas	(? <!--START UPDATE-->)(.*?)(?:Index Berita ya		2	
	2 1	Kompas	<A[^>]*?HREF<=\\s*[\"?](\\^>]+)[\"?]<		2	
	2 2	Kompas	<[Pp]>	□□	1	
	2 3	Kompas	(? (.?")(?&nb<		2	
	2 4	Kompas	<a href.*?>(.?")<		2	
	3 2	Kompas	<[^>]^>		1	
	4 2	Kompas	□		1	
	5 2	Kompas	□		1	
	6 2	Kompas	□		1	
	7 2	Kompas	□		1	
	1 1	nama	123		2	
	1 1	Surya	(?=+<		2	
	1 2	Surya	</?b>		1	
	1 3	Surya	□		1	
	1 4	Surya	(?&font class=judul-artikel.*?>(.?")<		2	
	2 2	Surya	</?i>		1	

Gambar 3.13 Tabel NewsRegex

3.8.4 Tabel LinksIndex

Tabel ini berisi data-data tentang berita itu sendiri, yaitu judul berita, waktu berita, dan isi berita. Atribut-atributnya adalah :

NPName : nama penyedia berita

PDName : nama direktori

LIURL : link isi berita

LIContent : isi berita

WktBerita : waktu berita

WktMasuk: waktu berita masuk aplikasi

LJudul : judul berita

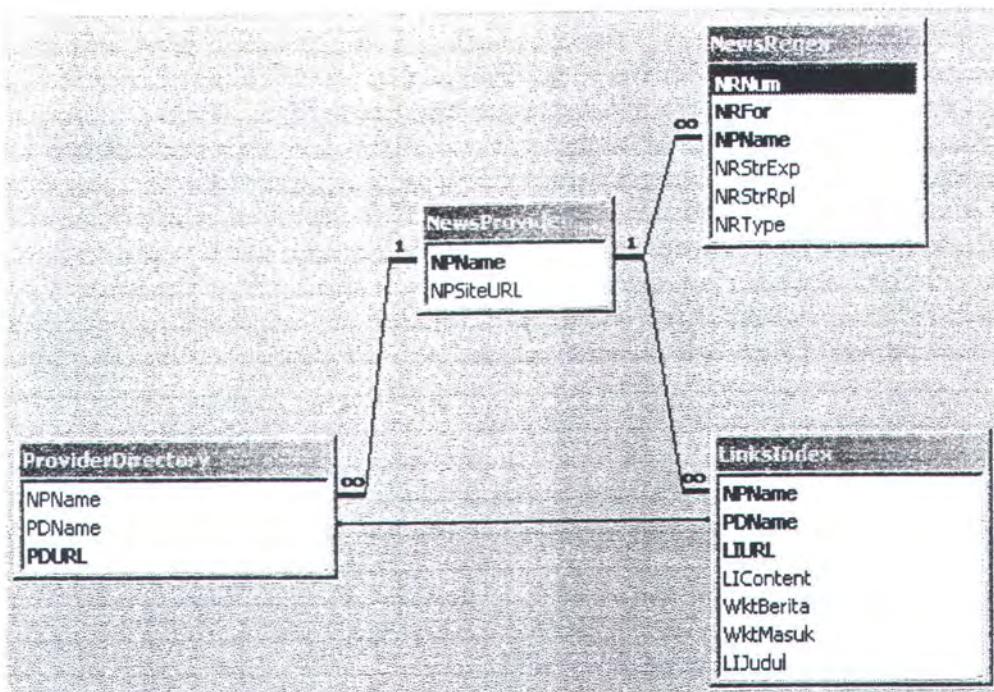
LinksIndex : Table			
NPName	PDName	LiURL	LJudul
Detik	Sports	http://www.detiksport.com/sepakbola/dunia/2003/07/08/20030708-144214.shtml	
Detik	news	http://www.detik.com/peristiwa/2003/07/08/20030708-142824.shtml	
Detik	news	http://www.detik.com/peristiwa/2003/07/08/20030708-142857.shtml	
Detik	news	http://www.detik.com/peristiwa/2003/07/08/20030708-143023.shtml	
Detik	news	http://www.detik.com/peristiwa/2003/07/08/20030708-143309.shtml	
Detik	Finance	http://www.detik.com/bisnis/keuangan/2003/07/08/20030708-142135.shtml	
Detik	Finance	http://www.detik.com/bisnis/ekonomi/2003/07/08/20030708-135030.shtml	
Detik	Finance	http://www.detik.com/bisnis/ekonomi/2003/07/08/20030708-142349.shtml	
Detik	Finance	http://www.detik.com/bisnis/ekonomi/2003/07/08/20030708-144409.shtml	
Detik	Sports	http://www.detiksport.com/sepakbola/spanyol/2003/07/08/20030708-125634.shtml	
Detik	news	http://www.detik.com/peristiwa/2003/07/08/20030708-121100.shtml	
Detik	news	http://www.detik.com/peristiwa/2003/07/08/20030708-122330.shtml	
Detik	news	http://www.detik.com/peristiwa/2003/07/08/20030708-122741.shtml	
Detik	news	http://www.detik.com/peristiwa/2003/07/08/20030708-123658.shtml	
Detik	news	http://www.detik.com/peristiwa/2003/07/08/20030708-124429.shtml	

LinksIndex : Table			
LiContent	WktBerita	WktMasuk	LJudul
detiksport - Jakarta, Top skore	08/07/2003 14:43:00	08/07/2003 15:16:04	Sukur Kembali ke Galatasaray
detikcom - Jakarta, Panglima T	08/07/2003 14:28:00	08/07/2003 15:12:37	Panglima TNI Beberkan Kronologis Pemb
detikcom - Jakarta, DPP PDIP	08/07/2003 14:29:00	08/07/2003 15:10:17	DPP PDIP Belum Ambil Putusan Soal No
detikcom - Jakarta, Kepala bac	08/07/2003 14:30:00	08/07/2003 15:08:54	PBB Desak Iran Izinkan Inspeksi atas Pro
detikcom - Jakarta, Tim dokter	08/07/2003 14:33:00	08/07/2003 15:07:18	Berhasil Dipisahkan, Kondisi Kembal Siai
detikcom - Jakarta, Pemerintah	08/07/2003 14:22:00	08/07/2003 14:56:52	Pemerintah Diminta Naikkan Nilai Buku B
detikcom - Jakarta, Unocal dar	08/07/2003 13:51:00	08/07/2003 14:52:09	Unocal-PLN Sepakati Harga Buy Out PLT
detikcom - Jakarta, Satu lagi s	08/07/2003 14:24:00	08/07/2003 14:49:36	Okttober 2003, Pacific TV Mulai Beroperas
detikcom - Jakarta, Indonesia c	08/07/2003 14:44:00	08/07/2003 14:47:48	Diyakini, Rating Utang Indonesia akan Me
Detiksport - Jakarta, Daya tarik	08/07/2003 12:57:00	08/07/2003 14:32:25	Beckham, Bintangi Iklan di Spanyol
detikcom - Jakarta, Panglima T	08/07/2003 12:12:00	08/07/2003 14:29:15	Panglima TNI: Ersa Disandera atau Meny
detikcom - Jakarta, Pembina k	08/07/2003 12:24:00	08/07/2003 14:28:11	Muhammadiyah: Kupon Asuransi Kematia
detikcom - Jakarta, PKB Mator	08/07/2003 12:28:00	08/07/2003 14:26:35	Dekah Verifikasi PKB Alwi Shihab
detikcom - Jakarta, Mungkink	08/07/2003 12:39:00	08/07/2003 14:24:20	Mungkinkah Menghadirkan Al Faruq di Si
detikcom - Jakarta, Ketua Umu	08/07/2003 12:44:00	08/07/2003 14:23:24	Keracunan, Hakim Pikir-Pikir untuk Banta

Gambar 3.14 Tabel LinksIndex

3.8.5 Relationship

Hubungan antar tabel (*relationship*) dalam database Microsoft Access ditunjukkan pada gambar 3.16.



Gambar 3.15 Relationship



BAB IV

IMPLEMENTASI PERANGKAT LUNAK

BAB 4

IMPLEMENTASI PERANGKAT LUNAK

Setelah dilakukan pemodelan dan perancangan sistem pada bab 3 diatas, selanjutnya dilakukan pembuatan aplikasi berdasarkan rancangan tersebut.

4.1 Implementasi Rancangan Sistem

Untuk mengimplementasikan rancangan sistem *client-server* ini, digunakan tools Microsoft Visual Studio .NET dalam membuat *web service*. Bahasa pemrograman yang digunakan adalah VB.NET. Web service tidak menghasilkan bentuk antarmuka melainkan hanya API bagi sistem lainnya. Deskripsi API ini dijelaskan dalam bentuk WSDL. *Web service* ini dirancang dengan tujuan sebagai jembatan antara *Client* dengan *Server*. Metode lain yang biasa dipakai adalah *remoting/socket*.

Di dalam *web service* ini mengandung class-class dan fungsi-fungsi utama dari aplikasi *Administrator* dan *Client*. Class-class tersebut adalah:

```
Public Class NewsProvider
    Public NPName As String
    Public NPSiteURL As String
    Public NPRegex As ArrayList           ' of newsregex
    Public NPDirecory As ArrayList        ' of provider directory
End Class

Public Class NewsRegex
    Public NRNum As String
    Public NPName As String
    Public NRFor As String
    Public NRStrExp As String
    Public NRStrRpl As String
    Public NRTypr As String
End Class
```

Class

```

Public Class ProviderDirectory
    Public NPName As String
    Public PDName As String
    Public PDURL As String
End Class

Public Class LinksIndex
    Public NPName As String
    Public PDName As String
    Public LIURL As String
    Public LJJudul As String
    Public LIContent As String
    Public WktMasuk As String
    Public WktBerita As String
End Class

```

Aplikasi *Administrator* menggunakan *web service* untuk memanggil fungsi-fungsi untuk memanipulasi tabel-tabel dalam *database*, yaitu tabel *NewsProvider*, *ProviderDirectory*, dan *NewsRegex*. Sedangkan aplikasi *Client* menggunakan *web service* untuk mengambil isi berita secara *offline* dari *database* untuk dibaca oleh *client/pengguna*.

4.2 Perancangan Aplikasi *Administrator*

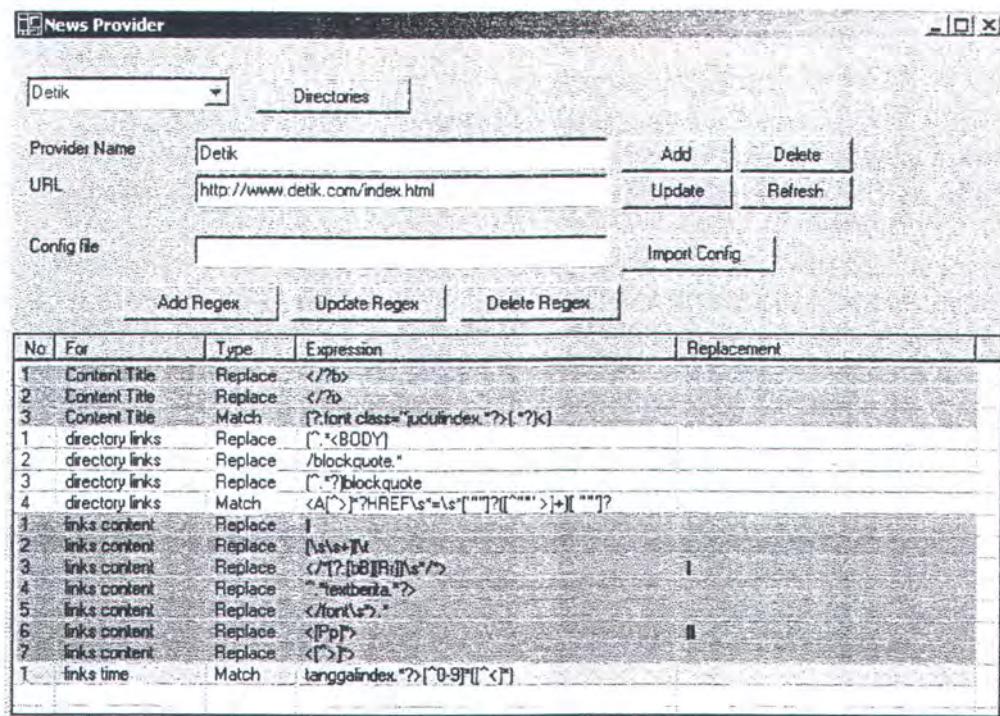
Aplikasi *Administrator* dibuat untuk memanipulasi *database*, khususnya untuk *insert*, *delete* dan *update* tabel-tabel penting yaitu *NewsProvider*, *ProviderDirectory*, dan *NewsRegex*. Untuk tabel *NewsProvider*, *Administrator* dapat menambah, mengubah/update dan menghapus salah satu penyedia berita (*news provider*). Untuk tabel *ProviderDirectory*, *Administrator* dapat menambah, mengubah/update dan menghapus salah satu direktori dari penyedia berita tertentu. Misalnya menghapus direktori News dari Detik, atau mengubah URL direktori Sports dari Detik.

Sedangkan yang terakhir untuk tabel *NewsRegex*, *Administrator* aplikasi dapat meng-*import file* konfigurasi *regex* dari tiap penyedia berita. Sedangkan *administrator regex* dapat menambah, mengubah/update dan menghapus salah satu formula *Regex (regular expression)* dari penyedia berita tertentu baik melalui aplikasi *client* ini, maupun langsung mengubah dari file konfigurasinya. Perlu diketahui bahwa satu penyedia berita memiliki beberapa *regex* yang sama untuk semua direktori dari penyedia berita tersebut.

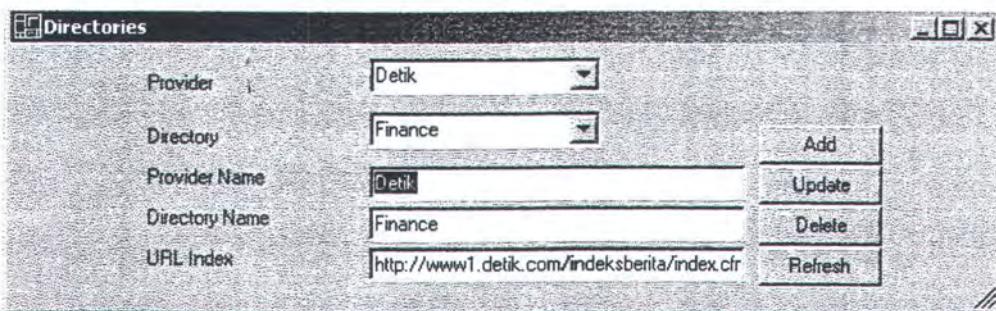
Tampilan awal dari aplikasi *Administrator* ditunjukkan pada gambar 4.1.

Aplikasi administrator memiliki beberapa fitur sebagai berikut: Tombol *directories* untuk memanipulasi direktori-direktori dari suatu penyedia berita. *Form* yang muncul ditunjukkan pada gambar 4.2.

Di samping itu ada tombol “*Import Config*” untuk mengambil *file* konfigurasi dari penyedia berita yang terletak pada *path* tertentu. Proses ini ditunjukkan pada gambar 4.3.



Gambar 4.1 Tampilan Awal Aplikasi News Administrator



Gambar 4.2 Form Directories

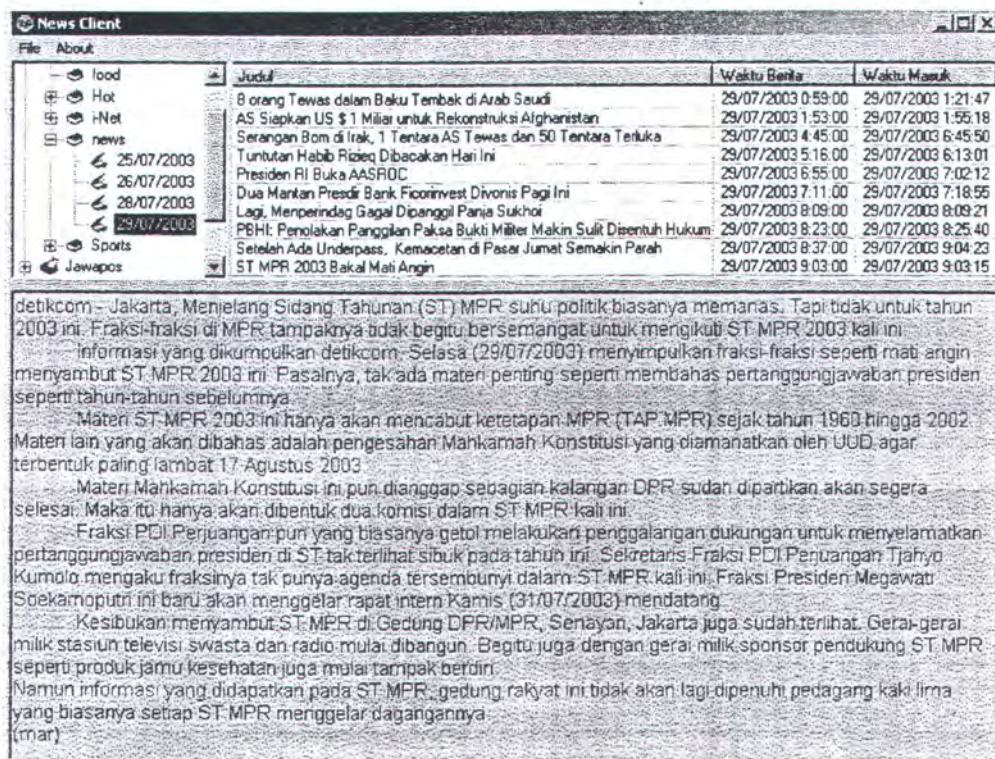


Gambar 4.3 Import Config

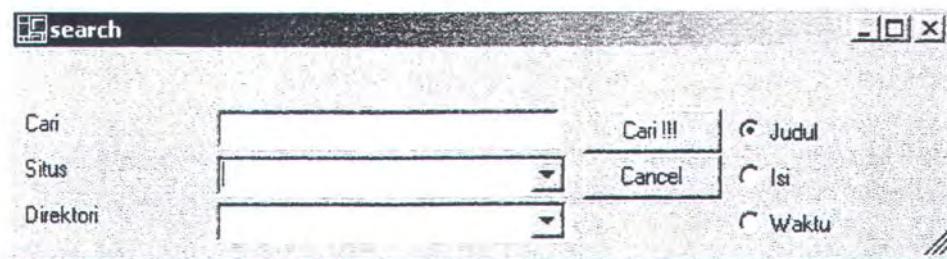
4.3 Perancangan Aplikasi *Client*

Aplikasi *Client* dibuat menyerupai Microsoft Outlook (*MS.Outlook-like*) untuk memudahkan interaksi dengan *client/pengguna*. Tampilan awal aplikasi ini ditunjukkan pada gambar 4.4.

Aplikasi *Client* ini memiliki fitur sebagai berikut: *TreeView* di menu sebelah kiri menunjukkan penyedia-penyedia berita dengan direktoriya masing-masing dan dikelompokkan berdasarkan tanggal berita. Sedangkan *ListView* di sebelah kanannya menunjukkan judul berita beserta waktu berita dan waktu masuk ke databasenya. Area utama di bawah menampilkan isi (*main content*) dari judul berita yang dipilih. Pada aplikasi ini juga bisa dilakukan pencarian berita dengan mengakses menu *file-search*. Form yang muncul ditunjukkan pada gambar 4.5.



Gambar 4.4 Tampilan Awal Aplikasi *Client*



Gambar 4.5 Form Search untuk Mencari Berita

Field-field tersebut berturut-turut untuk memasukkan *keyword* (kata kunci) yang dicari, situs penyedia berita, dan direktorinya. Kemudian di sisi kanan opsi-opsi untuk menentukan berita yang akan dicari *keyword*-nya pada judul, isi atau waktu.

4.4 Pembuatan Windows Service

Aplikasi ini bekerja sebagai proses *background* yang akan melakukan *looping* untuk mengambil berita secara berkala. *Looping* dilakukan dengan mengambil semua direktori dari semua penyedia berita (*provider*) yang ada dalam *database*. Proses ini terjadi secara terus menerus, sehingga apabila *looping* telah mencapai direktori terakhir dari penyedia terakhir, ia akan kembali ke penyedia berita dan direktori awal untuk mengecek lagi apakah ada berita (*link*) baru yang masuk ke situs penyedia berita tersebut. *Service* ini akan mengecek terus apakah ada berita baru yang masuk sesuai dengan penyedia berita saat proses *looping* tersebut terjadi. Berikut ini adalah *pseudo code* untuk fungsi looping tersebut:

```

Private NewsGathering()
arrprovider = getProviders()                                (ambil semua provider)
While Me.Running
    For Each provider In arrprovider                      (loop untuk tiap provider)
        For Each rgxprov In provider.NPRegex               (ambil semua Regex)
            Select Case rgxprov.NRFor
                Case "1"          arrrgxlink.Add(rgxprov)      (ambil Regex untuk link)
                Case "2"          arrrgxcontent.Add(rgxprov)   (ambil Regex untuk content)
                Case "3"          arrrgxtime.Add(rgxprov)     (ambil Regex untuk time)
                Case "4"          arrrgxtitle.Add(rgxprov)    (ambil Regex untuk title)
            End Select
            For Each provdir In provider.NPDirectory        (loop untuk tiap direktori)
                strRoot = getProvider(provdir.NPName).NPsiteURL
                streamweb = getHTMLContent(provdir.PDURL)
                strWeb = streamweb.ReadToEnd()

                arrlink = ParseHTML(strWeb, arrrgxlink)         (ambil array link)
                arrttitle = ParseHTML(strWeb, arrrgxtitle)       (ambil array title)
                arrtime = ParseHTML(strWeb, arrrgxtime)          (ambil array time)

                If arrtime.Count <= 0 Or arrlink.Count <> arrttitle.Count Then
                    strMessage += "count of links, title, time are not equal ..., please fix your regex"
                    strMessage += "links will not be added to database..."
                Else
                    strMessage += "add links to database..."
                    While i=1 < arrlink.Count (loop untuk ambil dan masukkan semua link ke database)
                        (cek link, jika link blm ada di database maka lakukan proses di bawah)

                            (masukkan data-data ke Class LinkIndex (linknya) )
                            linknya.NPName = provider.NPName
                            linknya.PDName = provdir.PDName
                            linknya.LIJudul = arrttitle.Item(i)
                            linknya.LIURL = arrlink.Item(i)
                            linknya.WktBerita = Now.Parse(arrtime.Item(i)).ToString
                            linknya.WktMasuk = Now.ToString

                            (ambil contents)
                            streamweb = getHTMLContent(linknya.LIURL)
                            strWeb = streamweb.ReadToEnd()
                            linknya.LIContent = ParseContent(strWeb, arrrgxcontent)

                            (masukkan array class LinkIndex (linknya) ke database)
                            addLink(linknya)
                            i ++

```

Dalam *service* ini terdapat pula fungsi untuk melakukan *web request* yaitu mengambil source HTML pada suatu situs. Fungsi lengkapnya adalah sebagai berikut:

```

Private Function getHTMLContent (surl As String) As Stream
    URLReq = WebRequest.Create(surl)
    URLReq.Proxy = New WebProxy("202.155.84.180", 8080)

    nc.UserName = "umum"
    nc.Password = "umum"
    URLReq.Proxy.Credentials = nc
    URLReq.Timeout = 60000
    URLRes = URLReq.GetResponse()
    Return URLRes.GetResponseStream()
End Function

```

Dalam service ini juga mengandung fungsi yang terpenting dari keseluruhan proses, yaitu fungsi *parsing HTML* untuk mengambil *link*, *time* (waktu berita), *title* (judul berita) dan fungsi untuk mengambil *content*. *Pseudo code* dari fungsi tersebut adalah sebagai berikut:

```

Function ParseHTML(cnt As String, arrRegex As ArrayList, Optional unique As Boolean = True) As ArrayList
    rRegEx As Regex
    mMatch As Match
    aMatch As New ArrayList()
    nr As newswebservice.NewsRegex
    strRet = cnt
    For Each nr In arrRegex
        If ismatch = True Then strRet = mMatch.ToString()
        If nr.NRStrExp <> Nothing Then strpattern = nr.NRStrExp
        If nr.NRTType = "1" Then
            If nr.NRStrRpl <> Nothing Then strrepl = nr.NRStrRpl
            strRet = rRegEx.Replace(strRet, strpattern, strrepl)
        Else
            mMatch = rRegEx.Match(strRet, strpattern)
            ismatch = True
        End If
    Next

```

```

If ismatch Then
    While mMatch.Success                                (loop untuk group matching)
        sMatch = mMatch.Groups(1).ToString
        aMatch.Add(sMatch)
        mMatch = mMatch.NextMatch()
    End While
End If
Return aMatch                                         (return array match)
End Function

```

Untuk fungsi *parsing* direktori *pseudo-code*-nya hampir sama dengan fungsi *parsing HTML* di atas, hanya saja yang di *parsing* adalah *content*-nya.

4.5 Perancangan Skenario *Regular expression*

Pada tahap ini dilakukan perancangan skenario, yaitu dalam bentuk formula *regex* (*regular expression*). Ada beberapa langkah yang harus dilakukan sebelum merancang skenario ini, yaitu sebagai berikut:

4.5.1 Pengenalan situs

Dalam tahap ini dilakukan pengenalan pola situs, mulai dari struktur situs, penamaan direktori, URL, dan lain-lain yang diperlukan bagi perancangan *regex* nantinya. Untuk mengenali suatu situs harus didapatkan *source HTML* dari situs tersebut. Umumnya diambil file yang mengandung indeks berita dari situs yang bersangkutan. Dari *source* ini kemudian diambil pola dengan melihat *tag-tag* khusus yang menentukan posisi *link* indeks berita, waktu berita, judul bahkan isi berita itu sendiri.

4.5.2 Pembuatan *regex* untuk *link*, *time* dan *title*

Setelah mengetahui pola situs kita dapat membuat formula *regex* untuk memfilter aspek-aspek yang kita perlukan. Yang paling penting di sini adalah *filtering* terhadap *link* indeks berita, karena *link-link* inilah yang nantinya akan merujuk kepada isi/content beritanya. Untuk dapat mengambil *link-link* yang sesuai, harus diketahui dimana posisi mulai dan akhir dari *link-link* yang akan diambil. Selain *link-link* tersebut, dilakukan pula *filtering* untuk mendapatkan waktu berita (*time*) dan judul berita (*title*) juga dengan *regex* tertentu yang dibuat berdasarkan pengenalan pola situs yang telah dilakukan sebelumnya.

4.5.3 Pembuatan *regex* untuk content

Setelah mendapatkan *link-link* indeks berita yang akan diambil, dilakukan pengambilan content/isi berita juga dengan *regex* hasil pengenalan pola dari *link* yang merujuk ke berita utama tersebut. Tahapan perancangan skenario ini akan lebih jelas diterangkan beserta contohnya pada bab uji coba (Bab 5).



BAB V

UJI COBA DAN ANALISA HASIL

BAB 5

UJI COBA DAN ANALISA HASIL

Pada bab ini akan dilakukan uji coba terhadap proses pengambilan berita. Situs-situs berita yang akan digunakan sebagai analisa meliputi detik (www.detik.com), republika (www.republika.co.id) dan kompas (www.kompas.co.id). Analisa ini dilakukan untuk mengetahui apakah proses pengambilan berita berjalan dengan lancar dengan menghasilkan *content* yang benar dan sesuai untuk tiap-tiap direktori dari situs berita tersebut. Uji coba juga dilakukan untuk menguji apakah proses *client-server* dapat berjalan dengan benar dan akurat.

5.1 Lingkungan Uji Coba

5.1.1 *Server*

Komputer yang dijadikan sebagai *server* mempunyai spesifikasi sebagai berikut :

Nama Komputer : LP-39

IP Address : 10.126.11.139

Prosesor : Pentium II 450 MHz

Memori fisik : 392 MB

Web Server : Microsoft Internet Information Service 5.0

Sistem Operasi : Windows 2000 Professional

5.1.2 *Client*

Komputer yang dijadikan sebagai *client* mempunyai spesifikasi sebagai berikut :

Nama Komputer : LP-39

IP Address : 10.126.11.139

Prosesor : Pentium II 450 MHz

Memori fisik : 392 MB

Sistem Operasi : Windows 2000 Professional

5.2 Skenario Uji Coba

Pada uji coba ini dilakukan perancangan skenario *regex* dan pengambilan berita terhadap tiga situs berita yang dianggap mampu merepresentasikan situs-situs berita di Indonesia. Situs-situs tersebut adalah detik , kompas dan republika. Pengambilan berita ini sebenarnya dilakukan oleh *server* secara berkala dan dalam bentuk proses background. Dalam proses background tersebut aplikasi akan mengambil *source HTML* dari situs berita. Untuk masing-masing direktori diambil *source*-nya untuk dilakukan filtering dengan *regular expression (regex)*. Kemudian untuk masing-masing direktori tersebut diambil *link-link* beritanya, judul dan waktunya. Terutama *link-link* tersebut sangat penting untuk mengambil *main content*-nya. Namun dalam uji coba ini akan menggunakan fungsi di *web service* untuk dapat memantau keberhasilan mengambil informasi berita per

penyedia situs. Sebab sebagai proses background, proses berjalan looping untuk semua situs, jadi tidak dapat dipantau satu persatu.

Skenario uji coba yang lain adalah dengan membandingkan antara *browsing* ke situs secara langsung dibandingkan dengan memakai aplikasi ini. Tujuannya adalah membandingkan besar/ukuran *file* yang dihasilkan. Uji coba juga ini dapat dijadikan referensi untuk mengetahui bahwa aplikasi ini dapat menghemat *bandwidth*.

5.2.1 Uji Coba pada Situs Detik

Uji coba ini dilakukan pada situs detik (www.detik.com) dengan direktori-direktori *Finance, food, Hot, i-Net, news, dan Sports*. *Regex* dirancang untuk dapat mengambil *content* dari tiap direktori dan dimasukkan ke *database*.

5.2.1.1 Perancangan skenario *regex*

Pertama akan dilakukan pengenalan pola terhadap situs detik ini, yaitu dengan mengambil *source HTML*-nya untuk indeks beritanya (<http://www.detik.com/index.html>). Setelah itu dapat diketahui bahwa *link-link* indeks berita dari situs detik berada diantara *tag* khusus, yaitu *<blockquote>*. Pengambilan *link* dilakukan dengan menghilangkan semua karakter dari awal sampai sebelum *blockquote*, *regex*-nya adalah sebagai berikut: “(^.*?)*blockquote*”. Kemudian menghilangkan semua karakter dari penutup tag *blockquote* (*</blockquote>*) sampai akhir karakter (habis), *regex*-nya adalah sebagai berikut: “/*blockquote*.*”. Kedua hal ini dapat dilakukan secara vise versa atau bisa dibolak-balik. Setelah mendapatkan daerah/posisi *link* indeks berita yang

akan diambil kemudian *link-link* tersebut diambil, yaitu dengan mengenali tag untuk *link* (`<href>`). *Regex*-nya adalah sebagai berikut: “`<A[^>]*?HREF\s*=\s*[""]*?([^\"]*>|+)[\s*"]*?`”. *Regex*-*regex* ini disimpan dalam tabel NewsRegex dengan urutan, fungsi dan tipe masing-masing seperti yang telah dijelaskan dalam bab 3.

5.2.1.2 Pengambilan berita

Data-data untuk situs detik seperti pada gambar 5.1 sebagai berikut:

Detik		http://www.detik.com/index.html	PDURL
	PDName		
+	Sports	http://www1.detik.com/indeksberita/index.cfm?fuseaction=index.detikSports	
+	news	http://www1.detik.com/indeksberita/index.cfm?fuseaction=index.detikNews	
+	i-Net	http://www1.detik.com/indeksberita/index.cfm?fuseaction=index.detiki-Net	
+	Hot	http://www1.detik.com/indeksberita/index.cfm?fuseaction=index.detikHot	
+	food	http://www1.detik.com/indeksberita/index.cfm?fuseaction=index.detikFood	
+	Finance	http://www1.detik.com/indeksberita/index.cfm?fuseaction=index.detikFinance	
*			

Gambar 5.1 Data Direktori Detik

Data-data *regex* untuk situs detik ditunjukkan pada gambar 5.2 sebagai berikut:

NRNum	NRF or	NPName	NRStrExp	NRStrRpl	NRType
1.1	Detik	(^<BODY>)		1	
2.1	Detik	/blockquote.*		1	
3.1	Detik	(^.*?)blockquote		1	
4.1	Detik	<A[^>]*?HREF\s*=\s*[""]*?([^\"]*> +)[\s*"]*?		2	
1.2	Detik	□		1	
2.2	Detik	(\s*+) \n		1	
3.2	Detik	</^([B][R])\s*/>	□	1	
4.2	Detik	^.*textberita.*?		1	
5.2	Detik	</font\s*>.*		1	
6.2	Detik	<[Pp]>	□□	1	
7.2	Detik	<[^>]*>		1	
1.3	Detik	tanggalindex.*?>[0-9]*(< >)		2	
1.4	Detik	</?b>		1	
2.4	Detik	</?i>		1	
3.4	Detik	(?i:font class="judulindex.*?>(*?)<)		2	

Gambar 5.2 Data Regex Detik

Setelah *web service* di-run, fungsi *parseDirectory* dipanggil, kemudian parameter-parameter dimasukkan, yaitu situs penyedia berita, direktori dan *includeContent* (untuk pilihan termasuk *content/tidak*) yang akan diambil *main content* beritanya. Seperti gambar 5.3 di bawah, uji coba yang dilakukan adalah mengambil berita dari situs detik (www.detik.com) dengan direktori *News*.

Parameter	Value
NPName:	<input type="text" value="Detik"/>
PDName:	<input type="text" value="News"/>
includeContent:	<input checked="" type="checkbox"/> true
<input type="button" value="Invoke"/>	

Gambar 5.3 Fungsi *ParseDirectory*

Setelah di-*invoke* untuk mendapatkan hasil pengambilan berita dalam bentuk XML. Fungsi ini akan mengembalikan nilai *true* jika berhasil mengambil informasi berita dan masuk ke *database*, dan *false* jika sebaliknya. Output hasil fungsi *parseDirectory* dalam format XML ditunjukkan pada gambar 5.4.

Setelah fungsi ini selesai, maka informasi berita telah berhasil masuk ke *database*, yaitu ke tabel *LinksIndex* untuk ditampilkan ke *client*. Gambar 5.5 menunjukkan data-data dari situs detik setelah masuk ke dalam *database*.

Dari gambar 5.5 tersebut dapat disimpulkan bahwa pengambilan informasi berita telah berhasil dilakukan dan dimasukkan ke *database*. Untuk selanjutnya, *client* akan mengakses berita-berita ini secara *offline*, atau langsung

mengambilnya dari *database*. Gambar 5.6 menunjukkan aplikasi *Client* membaca berita.

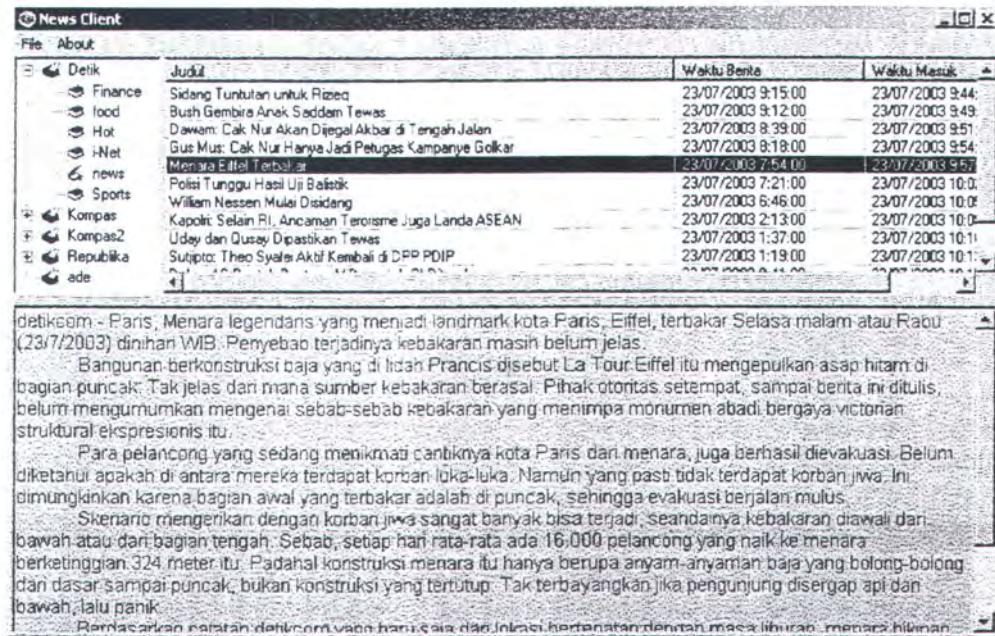
```
<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfAnyType xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://tempuri.org/">
  <anyType xsi:type="xsd:string">http://www.detik.com/peristiwa/2003/07/23/20030723-091529.shtml</anyType>
  <anyType xsi:type="xsd:string">http://www.detik.com/peristiwa/2003/07/23/20030723-091035.shtml</anyType>
  <anyType xsi:type="xsd:string">http://www.detik.com/peristiwa/2003/07/23/20030723-083816.shtml</anyType>
  <anyType xsi:type="xsd:string">http://www.detik.com/peristiwa/2003/07/23/20030723-091818.shtml</anyType>
  <anyType xsi:type="xsd:string">http://www.detik.com/peristiwa/2003/07/23/20030723-075228.shtml</anyType>
  <anyType xsi:type="xsd:string">http://www.detik.com/peristiwa/2003/07/23/20030723-072015.shtml</anyType>
  <anyType xsi:type="xsd:string">http://www.detik.com/peristiwa/2003/07/23/20030723-064507.shtml</anyType>
  <anyType xsi:type="xsd:string">http://www.detik.com/peristiwa/2003/07/23/20030723-021320.shtml</anyType>
  <anyType xsi:type="xsd:string">http://www.detik.com/peristiwa/2003/07/23/20030723-013647.shtml</anyType>
  <anyType xsi:type="xsd:string">http://www.detik.com/peristiwa/2003/07/23/20030723-011758.shtml</anyType>
  <anyType xsi:type="xsd:string">http://www.detik.com/peristiwa/2003/07/23/20030723-004051.shtml</anyType>
  <anyType xsi:type="xsd:string">http://www.detik.com/peristiwa/2003/07/23/20030723-000800.shtml</anyType>
</ArrayOfAnyType>
```

Gambar 5.4 Output Fungsi *parseDirectory* dalam format XML

NPName	PDName	UJURL	UJContent
Detik	News	http://www.detik.com/peristiwa/2003/07/23/20030723-000800.shtml	detikcom - Yogyakarta, Meski
Detik	News	http://www.detik.com/peristiwa/2003/07/23/20030723-004051.shtml	detikcom - Jakarta, Duta Besa
Detik	News	http://www.detik.com/peristiwa/2003/07/23/20030723-011758.shtml	detikcom - Solo, Keinginan The
Detik	News	http://www.detik.com/peristiwa/2003/07/23/20030723-013647.shtml	detikcom - Jakarta, Setelah be
Detik	News	http://www.detik.com/peristiwa/2003/07/23/20030723-021320.shtml	detikcom - Jakarta, Kapolri Jer
Detik	News	http://www.detik.com/peristiwa/2003/07/23/20030723-064507.shtml	detikcom - Banda Aceh, Menu
Detik	News	http://www.detik.com/peristiwa/2003/07/23/20030723-072015.shtml	detikcom - Jakarta, Polda Metr
Detik	News	http://www.detik.com/peristiwa/2003/07/23/20030723-075228.shtml	detikcom - Paris, Menara leger
Detik	News	http://www.detik.com/peristiwa/2003/07/23/20030723-081818.shtml	detikcom - Jakarta, Budayawan
Detik	News	http://www.detik.com/peristiwa/2003/07/23/20030723-083816.shtml	detikcom - Jakarta, Menurut pe
Detik	News	http://www.detik.com/peristiwa/2003/07/23/20030723-091035.shtml	detikcom - Jakarta, Kabar tewa
Detik	News	http://www.detik.com/peristiwa/2003/07/23/20030723-091529.shtml	detikcom - Jakarta, Setelah be

WktBerita	WktMasuk	UJJudul
23/07/2003 0:09:00	23/07/2003 10:18:07	DPD Golkar Yogyakarta Tolak Calonkan Akbar Tandjung
23/07/2003 0:41:00	23/07/2003 10:15:44	Dubes AS Bantah Bantuan Militer untuk RI Ditunda
23/07/2003 1:19:00	23/07/2003 10:13:47	Sutjipto: Theo Syafei Aktif Kembali di DPP PDIP
23/07/2003 1:37:00	23/07/2003 10:10:56	Uday dan Qusay Dipastikan Tewas
23/07/2003 2:13:00	23/07/2003 10:08:50	Kapolri: Selain RI, Ancaman Terorisme Juga Landa ASEAN
23/07/2003 6:46:00	23/07/2003 10:05:33	William Nessen Mulai Disidang
23/07/2003 7:21:00	23/07/2003 10:02:21	Polisi Tunggu Hasil Uji Balistik
23/07/2003 7:54:00	23/07/2003 9:57:20	Menara Eiffel Terbakar
23/07/2003 8:18:00	23/07/2003 9:54:23	Gus Mus: Cak Nur Hanya Jadi Petugas Kampanye Golkar
23/07/2003 8:39:00	23/07/2003 9:51:57	Dawam: Cak Nur Akan Dijegal Akbar di Tengah Jalan
23/07/2003 9:12:00	23/07/2003 9:49:13	Bush Gembira Anak Saddam Tewas
23/07/2003 9:15:00	23/07/2003 9:44:13	Sidang Tuntutan untuk Rizieq

Gambar 5.5 Tabel setelah *link-link* dan berita masuk



Gambar 5.6 Aplikasi client-Melihat Isi Berita

5.2.2 Uji Coba pada Situs Republika

Uji coba ini dilakukan pada situs republika (www.republika.co.id) dengan direktori-direktori *Ekonomi Bisnis*, *Luar Negeri*, *Nasional*, *Olahraga*, dan *Topik Kita*. *Regex* dirancang untuk dapat mengambil *content* dari tiap direktori dan dimasukkan ke *database*.

5.2.2.1 Perancangan skenario *regex*

Seperti situs detik, dilakukan pengenalan pola terhadap situs republika ini, yaitu dengan mengambil *source HTML*-nya untuk indeks beritanya (http://www.republika.co.id/indeks_berita_online.asp?kat_id=23). Setelah itu dapat diketahui bahwa *link-link* indeks berita dari situs republika ini berada diantara *tag* khusus, yaitu <!-- Content: Delimiter --> <!-- Content: Tengah -->

> dengan karakter “*Hal: 1 |*”. Pola ini selalu terjadi pada situs republika ini, yaitu navigasi halaman, karena itulah karakter khusus ini dimanfaatkan untuk mengambil *link-link* berita. Pengambilan *link* dilakukan dengan mencocokkan (*Match*) semua karakter antara *tag* dan karakter khusus di atas “**Delimiter.*?Content: Tengah.*?>(.*)?(:Hal.*?1.*?\|)**”. Kemudian mengambil semua *link* yaitu dengan mencocokkan karakter setelah *tag href*. *Regex*-nya adalah sebagai berikut: “**<A[^>]*?HREF\s*=\s*[""]?([^\'"]*>]+[""]?**”.

5.2.2.2 Pengambilan berita

Data-data direktori beserta URL-nya untuk situs republika seperti pada gambar 5.7 sebagai berikut:

PDName	PDURL
* Ekonomi Bisnis	http://www.republika.co.id/index_berita_online.asp?kat_id=21
* Luar Negeri2	http://www.republika.co.id/index_berita_online.asp?kat_id=248
* Nasional	http://www.republika.co.id/index_berita_online.asp?kat_id=23
► * Olahraga2	http://www.republika.co.id/index_berita_online.asp?kat_id=247
* Topik Kita	http://www.republika.co.id/index_berita_online.asp?kat_id=156

Gambar 5.7 Data Direktori Republika

Data-data *regex* untuk situs republika ditunjukkan pada gambar 5.8 sebagai berikut:

NRNum	NRFord	NPName	NRStrExp	NRSIrrpt	NRTyp
1 1		Republika	Delimiter.*?Content: Tengah.*>(.*)?(?:Hal.:*?1.	2	
2 1		Republika	<A[^>]*?HREF\s*= s [" ? ?([^\n>]+) "]?	2	
1 2		Republika	□		1
2 2		Republika	□		1
3 2		Republika	□		1
4 2		Republika	□		1
5 2		Republika	\n\n]+		1
6 2		Republika	<!-- End Load Inisial Redaktur -->.*?>(.*)<hr	2	
7 2		Republika	 	□	1
8 2		Republika	<.*?>		1
9 2		Republika	(W)+ 		1
1 3		Republika	 		1
2 3		Republika	Delimiter.*?Content: Tengah.*>(.*)?(?:Hal.:*?1.	2	
3 3		Republika	<.*?>(?:<.*?>)*(.*)</	2	
1 4		Republika	</?([bu])>		1
2 4		Republika	Delimiter.*?Content: Tengah.*>(.*)?(?:Hal.:*?1.	2	
3 4		Republika	href.*?>.*?<.*?>(.*)</		2

Gambar 5.8 Data Regex Republika

Seperti pada situs detik, fungsi *parseDirectory* pada web service dijalankan, kemudian dimasukkan parameter-parameter masing-masing *NPName*, *PDName*, dan *includeContent*. Direktori yang akan diuji coba pada situs republika ini adalah direktori *Nasional*. Gambar 5.9 menunjukkan proses tersebut.

Parameter	Value
NPName:	Republika
PDName:	Nasional
includeContent:	true
Invoke	

Gambar 5.9 Fungsi *ParseDirectory*

Setelah di-*invoke* untuk mendapatkan hasil pengambilan berita dalam bentuk XML. Fungsi ini akan mengembalikan nilai *true* jika berhasil mengambil informasi berita dan masuk ke *database*, dan *false* jika sebaliknya. Output hasil



fungsi `parseDirectory` dalam format XML ditunjukkan pada gambar 5.10 di bawah ini:

```
<?xml version="1.0" encoding="utf-8" ?>
- <ArrayOfAnyType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://tempuri.org/"/>
  <anyType xsi:type="xsd:string">http://www.republika.co.id/online_detail.asp?
    id=132651&kat_id=23</anyType>
  <anyType xsi:type="xsd:string">http://www.republika.co.id/online_detail.asp?
    id=132621&kat_id=23</anyType>
  <anyType xsi:type="xsd:string">http://www.republika.co.id/online_detail.asp?
    id=132620&kat_id=23</anyType>
  <anyType xsi:type="xsd:string">http://www.republika.co.id/online_detail.asp?
    id=132619&kat_id=23</anyType>
  <anyType xsi:type="xsd:string">http://www.republika.co.id/online_detail.asp?
    id=132618&kat_id=23</anyType>
  <anyType xsi:type="xsd:string">http://www.republika.co.id/online_detail.asp?
    id=132617&kat_id=23</anyType>
  <anyType xsi:type="xsd:string">http://www.republika.co.id/online_detail.asp?
    id=132616&kat_id=23</anyType>
  <anyType xsi:type="xsd:string">http://www.republika.co.id/online_detail.asp?
    id=132612&kat_id=23</anyType>
  <anyType xsi:type="xsd:string">http://www.republika.co.id/online_detail.asp?
    id=132609&kat_id=23</anyType>
  <anyType xsi:type="xsd:string">http://www.republika.co.id/online_detail.asp?
    id=132608&kat_id=23</anyType>
</ArrayOfAnyType>
```

Gambar 5.10 Output Fungsi `parseDirectory` dalam format XML

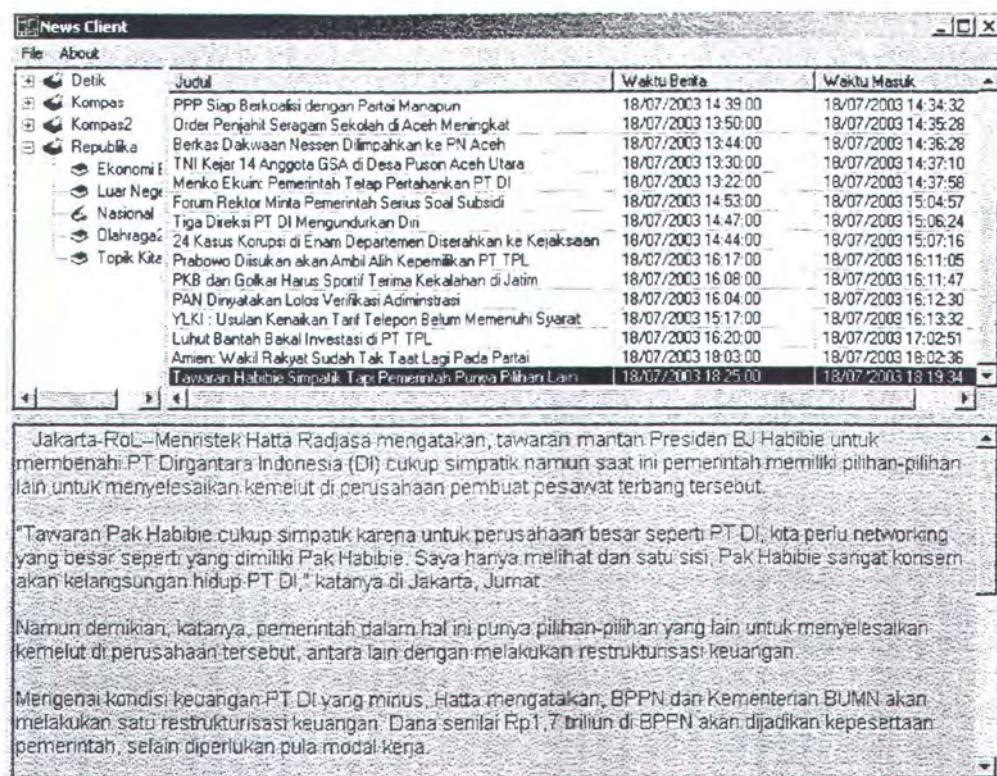
Setelah fungsi ini selesai, maka informasi berita telah berhasil masuk ke *database*, yaitu ke tabel *LinksIndex* untuk ditampilkan ke *client*. Gambar 5.11 menunjukkan data-data dari situs republika setelah masuk ke dalam *database*.

Dari gambar 5.11 dapat disimpulkan bahwa pengambilan informasi berita telah berhasil dilakukan dan dimasukkan ke *database*. Untuk selanjutnya, *client* akan mengakses berita-berita ini secara *offline*, atau langsung mengambilnya dari *database*. Gambar 5.12 menunjukkan aplikasi *Client* membaca berita.

NPName	PDName	LJURL	LJContent
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132580&kat_id=23	Jakarta-Ro-Polisi menemuk
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132540&kat_id=23	Bandung-Ro-Setelah dua h
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132526&kat_id=23	Jakarta-Ro-Departemen Da
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132523&kat_id=23	Manado-Ro-Gunung Soput
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132522&kat_id=23	Jakarta-Ro-Departemen Pe
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132475&kat_id=23	Jakarta-Ro-Pengamat ekor
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132474&kat_id=23	Jakarta-Ro-Ketua MPR Ar
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132473&kat_id=23	Jakarta-Ro-Rancangan Per
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132444&kat_id=23	Jakarta-Ro-Pemerintah me
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132433&kat_id=23	Jakarta-Ro-PT Jamsostek
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132608&kat_id=23	Jakarta-Ro-Ketua Umum P
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132591&kat_id=23	Banda Aceh-Ro-Memasuki
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132590&kat_id=23	Banda Aceh-Ro-Pihak Kej
Rapublika	Nasional	http://www.republika.co.id/online_detail.asp?id=132587&kat_id=23	Lhok Seumawe-Ro-TNI/Pol
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132586&kat_id=23	Jakarta-Ro-Menko Ekuin D
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132616&kat_id=23	Jakarta-Ro-Forum Rektor II
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132612&kat_id=23	Bandung-Ro-Tiga Direksi d
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132609&kat_id=23	Jakarta-Ro-Sebanyak 24 k
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132620&kat_id=23	Medan-Ro-Mantan Pangko
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132619&kat_id=23	Jakarta-Ro-Partai Kebangk
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132618&kat_id=23	Jakarta-Ro-Partai Amanat
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132617&kat_id=23	Jakarta-Ro-Yayasan Lemt
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132621&kat_id=23	Medan-Ro-Mantan Menpar
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132651&kat_id=23	Jakarta-Ro-Ketua Umum D
Republika	Nasional	http://www.republika.co.id/online_detail.asp?id=132652&kat_id=23	Jakarta-Ro-Menristek Hatt

WktBerita	WktMasuk	LJJudul
18/07/2003 11:46:00	18/07/2003 12:26:46	Tas Diduga Bom Ditemukan di Komplek Perumahan DPR
18/07/2003 10:43:00	18/07/2003 12:27:21	SPFKK PT DI Tak akan Halangi Karyawan yang Dapat Panggilan Bekerja Kembali
18/07/2003 10:17:00	18/07/2003 12:28:01	Pemerintah Segera Lantik Imam Utomo Sebagai Gubernur Jatim
18/07/2003 10:07:00	18/07/2003 12:28:46	Gunung Soputan Lontarkan Debu Panas Setinggi 1.000 Meter
18/07/2003 9:55:00	18/07/2003 12:29:33	Mulai 2004 Sekolah Jadi Pelaksana Ujian Akhir Nasional
17/07/2003 18:09:00	18/07/2003 12:30:16	Didik: Opsi PPM Langgar TAP MPR
17/07/2003 18:04:00	18/07/2003 12:31:02	Biaya Keamanan ST MPR 2003 Rp 1,1 Miliar
17/07/2003 18:00:00	18/07/2003 12:31:58	RPP Bantuan Dana Parpol Selesai Akhir Juli
17/07/2003 15:05:00	18/07/2003 12:32:33	Pemerintah Putuskan PT DI Panggil 3.000 Karyawan
17/07/2003 14:36:00	18/07/2003 12:33:01	Persiapan Dana Rp 80 M Untuk Antisipasi PHK di PT DI
18/07/2003 14:39:00	18/07/2003 14:34:32	PPP Siap Berkolalisi dengan Partai Manapun
18/07/2003 13:50:00	18/07/2003 14:35:28	Order Perajahit Seragam Sekolah di Aceh Meningkat
18/07/2003 13:44:00	18/07/2003 14:36:28	Berkas Dakwaan Nessan Diliimpahkan ke PN Aceh
18/07/2003 13:30:00	18/07/2003 14:37:10	TNI Kejar 14 Anggota GSA di Desa Puson Aceh Utara
18/07/2003 13:22:00	18/07/2003 14:37:58	Menko Ekuin: Pemerintah Tetap Pertahankan PT DI
18/07/2003 14:53:00	18/07/2003 15:04:57	Forum Rektor Minta Pemerintah Senus Soal Subsidi
18/07/2003 14:47:00	18/07/2003 15:06:24	Tiga Direksi PT DI Mengundurkan Diri
18/07/2003 14:44:00	18/07/2003 15:07:16	24 Kasus Korupsi di Enam Departemen Diserahkan ke Kejaksaan
18/07/2003 16:17:00	18/07/2003 16:11:05	Prabowo Diisukan akan Ambil Alih Kepemilikan PT TPL
18/07/2003 16:08:00	18/07/2003 16:11:47	PKB dan Golkar Harus Sportif Terima Kekalahannya di Jatim
18/07/2003 16:04:00	18/07/2003 16:12:30	PAN Dinyatakan Lulus Verifikasi Administrasi
18/07/2003 15:17:00	18/07/2003 16:13:32	YLKI : Usulan Kenaikan Tarif Telepon Belum Memenuhi Syarat
18/07/2003 16:20:00	18/07/2003 17:02:51	Luhut Bantah Bakal Investasi di PT TPL
18/07/2003 18:03:00	18/07/2003 18:02:36	Amien: Wakil Rakyat Sudah Tak Taat Lagi Pada Partai
18/07/2003 18:25:00	18/07/2003 18:19:34	Tawaran Habibie Simpatik Tapi Pemerintah Punya Pilihan Lain

Gambar 5.11 Tabel setelah link-link dan berita masuk



Gambar 5.12 Aplikasi client-Melihat Isi Berita

5.2.3 Uji Coba pada Situs Kompas

Uji coba ini dilakukan pada situs kompas (www.kompas.co.id) dengan direktori-direktori *Ekonomi*, *Gaya Hidup*, *Kesehatan*, *Teknologi*, *Metropolitan*, *Olah Raga*, dan *Topik Kita*. Regex dirancang untuk dapat mengambil *content* dari tiap direktori dan dimasukkan ke *database*. Skenario *regex* untuk kompas ini sengaja dibagi menjadi dua, yaitu kompas dengan direktori *Indeks*, dan kompas2 dengan direktori-direktori seperti diuraikan di atas. Hal ini dilakukan karena pola antara indeks berita utama kompas dengan direktori-direktori lainnya berlainan.

5.2.3.1 Perancangan skenario *regex*

Seperti situs-situs sebelumnya, dilakukan pengenalan pola terhadap situs kompas ini, yaitu dengan mengambil *source HTML*-nya untuk indeks berita utamanya (<http://www.kompas.co.id/utama/index.cfm>). Setelah itu dapat diketahui bahwa *link-link* indeks berita dari situs kompas ini berada diantara *tag* khusus, yaitu `<!--START UPDATE-->` dengan karakter “*Index Berita yang lalu*”. Pola ini selalu terjadi pada situs kompas ini, yaitu fitur untuk mencari indeks berita yang lalu, karena itulah karakter khusus ini dimanfaatkan untuk mengambil *link-link* berita. Pengambilan *link* dilakukan dengan mencocokkan (*Match*) semua karakter antara *tag* dan karakter khusus di atas “`(?:<!--START UPDATE-->)(.*?)(?:Index Berita yang lalu)`”. Kemudian mengambil semua *link* yaitu dengan mencocokkan karakter setelah *tag href*. *Regex*-nya adalah sebagai berikut:

“`<A[^>]*?HREF\s*=\\s*[""]*?([\""\">>]+)[\""]*`”.

Sedangkan untuk kompas2 dengan direktori-direktorinya menggunakan skenario *regex* yang berbeda karena pola yang berbeda. Pada kompas2 ini, direktori kesehatan dipakai sebagai contoh dengan indeksnya sebagai berikut: (<http://www.kompas.co.id/kesehatan/index.htm>). Dalam *source HTML*-nya, dapat dilihat bahwa *link-link* indeks berita dari direktori kesehatan ini ada setelah karakter khusus, yaitu “*last update*” dan sebelum “*<!--FLOATING SCRIPT*”. Pola ini selalu terjadi, bahwa *link-link* indeks berita pada direktori kesehatan ini selalu berada pada posisi tersebut. Begitu pula pada direktori-direktori lain, pola ini selalu terjadi. Pengambilan *link* dilakukan dengan mencocokkan (*match*) semua karakter dari setelah “*last update*” hingga sebelum

karakter “<!-- ----- FLOATING SCRIPT”, regex-nya adalah : ”last.*?update(.*)<..*?FLOATING SCRIPT”. Kemudian link-link tersebut diambil dengan regex “<A[^>]*?HREF\s*=|\s*[""]*?([^\"]+>]+htm)[“”]*?”.

5.2.3.2 Pengambilan berita

Data-data untuk situs kompas seperti pada gambar 5.13 sebagai berikut:

-		Kompas	http://www.kompas.co.id/	
		PDName	PDURL	
	+ Indeks		http://www.kompas.co.id/utama/index.cfm	
*				
-		Kompas2	http://www.kompas.co.id/	
		PDName	PDURL	
	+ Ekonomi		http://www.kompas.co.id/bisnis/bisnis.cfm	
	+ Gaya Hidup		http://www.kompas.co.id/gayahidup/index.htm	
	+ Kesehatan		http://www.kompas.co.id/kesehatan/index.htm	
►	+ Metropolitan		http://www.kompas.co.id/metro/index.htm	
	+ Olah Raga		http://www.kompas.co.id/olahraga/index.htm	
	+ Teknologi		http://www.kompas.co.id/teknologi/index.htm	
*				

Gambar 5.13 Data Direktori Kompas dan Kompas2

Data-data regex untuk situs kompas dan kompas2 ditunjukkan masing-masing pada gambar 5.14 dan gambar 5.15 sebagai berikut:

NRNum	NRFor	NPName	NRStrExp	NRStrRpl	NRTypE
1.1	Kompas		(?:<!--START UPDATE-->)(.*?)(?:Index Berita ya	2	
2.1	Kompas		<A[^>]*?HREF\s*= \s*[""]*?([^\"]+>]+htm)[“”]*?	2	
1.2	Kompas		<!--Start-->(.*?)<!--fotonews-->	2	
2.2	Kompas		<[Pp]>	□□	1
3.2	Kompas		<[^>]*>		1
4.2	Kompas		□		1
5.2	Kompas		□		1
6.2	Kompas		□		1
7.2	Kompas		□		1
8.2	Kompas		 		1
9.2	Kompas		[\\n\\r]+		1
1.3	Kompas		(?:<!--START UPDATE-->)(.*?)(?:Index Berita ya	2	
2.3	Kompas		(?:<i>)(.*?)(?:wib<)	2	
1.4	Kompas		(?:<!--START UPDATE-->)(.*?)(?:Index Berita ya	2	
2.4	Kompas		<a href.*?>(.*?)<	2	

Gambar 5.14 Data Regex Kompas

NRNum	NRFor	NPName	NRStrExp	NRStrRpl	NRTtype
1.1	Kompas2		last.*?update(?)<!.?FLOATING SCRIPT		2
2.1	Kompas2		<A[^>]*?HREF\s=\s[""]?([""]>)+htm["" ?		2
1.2	Kompas2		<!--Start-->(?)<--fotanews-->		2
2.2	Kompas2		<p>	□	1
3.2	Kompas2		<.?>		1
4.2	Kompas2		□		1
5.2	Kompas2		□		1
6.2	Kompas2		□		1
7.2	Kompas2		□		1
8.2	Kompas2		[nvr]+		1
9.2	Kompas2		 		1
1.3	Kompas2		(?:last.*?update(?<.?>+)(.*?)(?:wib)		2
1.4	Kompas2		(^.*?last.*?update)		1
2.4	Kompas2		(?:<A[^>]*?HREF\s=\s[""]?[^>]+htm["" ?>		2

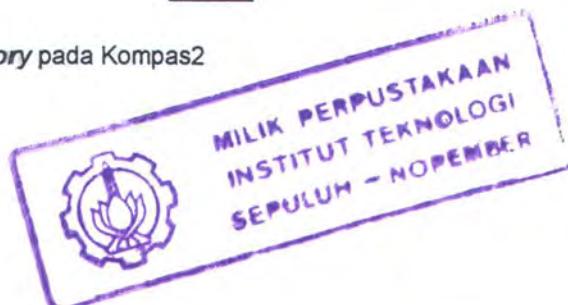
Gambar 5.15 Data Regex Kompas2

Seperti pada situs republika, fungsi *parseDirectory* pada web service dijalankan, kemudian dimasukkan parameter-parameter masing-masing *NPName*, *PDName*, dan *includeContent*. Direktori yang akan diuji coba pada situs kompas ini adalah direktori *Indeks*. Sedangkan untuk kompas2, direktori yang diuji coba adalah kesehatan. Gambar 5.16 dan gambar 5.17 masing-masing menunjukkan proses tersebut.

Parameter	Value
NPName:	Kompas
PDName:	Indeks
includeContent:	true
<input type="button" value="Invoke"/>	

Gambar 5.16 Fungsi ParseDirectory pada Kompas

Parameter	Value
NPName:	Kompas2
PDName:	Kesehatan
includeContent:	true
<input type="button" value="Invoke"/>	

Gambar 5.17 Fungsi ParseDirectory pada Kompas2

Setelah di-*invoke* untuk mendapatkan hasil pengambilan berita dalam bentuk XML. Fungsi ini akan mengembalikan nilai *true* jika berhasil mengambil informasi berita dan masuk ke *database*, dan *false* jika sebaliknya. Output hasil fungsi *parseDirectory* dalam format XML ditunjukkan pada gambar 5.18 untuk kompas, dan gambar 5.19 untuk kompas2 di bawah ini:

```
<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfAnyType xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://tempuri.org/">
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/161427.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/160833.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/155124.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/151915.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/145810.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/140505.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/140137.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/134125.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/133506.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/133354.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/125210.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/124240.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/123409.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/110925.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/101737.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/090309.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/085243.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/073624.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/042701.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/042009.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/040702.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/032206.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/021800.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/015625.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/011130.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/003159.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/161415.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/115542.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/113053.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/104213.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/100028.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/utama/news/0307/14/080627.htm</anyType>
</ArrayOfAnyType>
```

Gambar 5.18 Output *parseDirectory* dalam XML untuk Kompas

```
<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfAnyType xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://tempuri.org/">
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/kesehatan/news/0307/11/082859.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/kesehatan/news/0307/11/180409.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/kesehatan/news/0307/11/095205.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/kesehatan/news/0307/11/092010.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/kesehatan/news/0307/11/091243.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/kesehatan/news/0307/11/090805.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/kesehatan/news/0304/18/231436.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/kesehatan/news/0304/18/231005.htm</anyType>
  <anyType xsi:type="xsd:string">http://www.kompas.co.id/kesehatan/news/0304/18/230607.htm</anyType>
</ArrayOfAnyType>
```

Gambar 5.19 Output *parseDirectory* dalam XML untuk Kompas2

Setelah fungsi ini selesai, maka informasi berita telah berhasil masuk ke *database*, yaitu ke tabel *LinksIndex* untuk ditampilkan ke *client*. Gambar 5.20 untuk kompas, gambar 5.21 untuk kompas2 menunjukkan data-data dari situs-situs tersebut setelah masuk ke dalam *database*.

NPName	PDName	LURL	LiContent
Kompas	Indeks	http://www.kompas.co.id/utama/news/0307/14/080627.htm	Kompas/MH Samsul HadiDijac
Kompas	Indeks	http://www.kompas.co.id/utama/news/0307/14/100028.htm	Ribuan karyawan PT Dirgantara
Kompas	Indeks	http://www.kompas.co.id/utama/news/0307/14/104213.htm	Tiga Direktur PT Dirgantara Ind
Kompas	Indeks	http://www.kompas.co.id/utama/news/0307/14/115542.htm	Kapolda Metro Jaya Ijen (Pol)
Kompas	Indeks	http://www.kompas.co.id/utama/news/0307/14/161415.htm	Sebuah pesawat latih TNI AU j
Kompas	Indeks	http://www.kompas.co.id/utama/news/0307/14/113053.htm	Ilustrasi KCMKapolres Jakarta
Kompas	Indeks	http://www.kompas.co.id/utama/news/0307/14/011130.htm	Indonesia berupaya keras men
Kompas	Indeks	http://www.kompas.co.id/utama/news/0307/14/003159.htm	Ketua Badan Eksekutif Mahas
Kompas	Indeks	http://www.kompas.co.id/utama/news/0307/14/015625.htm	Pemerintah Jepang melalui bar
Kompas	Indeks	http://www.kompas.co.id/utama/news/0307/14/021800.htm	Mantan Menteri Kehutanan dan
Kompas	Indeks	http://www.kompas.co.id/utama/news/0307/14/032206.htm	Keadaan darurat militer di prop
Kompas	Indeks	http://www.kompas.co.id/utama/news/0307/14/040702.htm	Direktorat Jenderal (Dirjen) Pos
Kompas	Indeks	http://www.kompas.co.id/utama/news/0307/14/042009.htm	Satu skuadron pasukan Pakta
Kompas	Indeks	http://www.kompas.co.id/utama/news/0307/14/090309.htm	Mantan Kapala Staf Umum (Ka
Kompas	Indeks	http://www.kompas.co.id/utama/news/0307/14/085243.htm	Saham PT Bank Mandiri Tbk h
Kompas	Indeks	http://www.kompas.co.id/utama/news/0307/14/073624.htm	Sebuah kelompok yang mengk

WktBerita	WktMasuk	JJudit	
14/07/2003 8:05:00	19/07/2003 12:58:38	Karyawan Dilarang Memasuki Kompleks PT Dirgantara Indonesia	
14/07/2003 9:59:00	19/07/2003 12:58:37	Ribuan Karyawan PT DI Long March ke DPRD Jabar	
14/07/2003 10:40:00	19/07/2003 12:58:37	Tiga Direktur PT DI Tolak Kebijakan "Merumahkan" Karyawan	
14/07/2003 11:54:00	19/07/2003 12:58:36	Kapolda: Kekuatannya Berbahaya	
14/07/2003 16:13:00	19/07/2003 12:58:36	Pesawat Latih TNI AU Jatuh di Waduk Jatiluhur	
14/07/2003 11:29:00	19/07/2003 12:58:36	Jenis Ledakan di DPR/MPR Mirip dengan Ledakan di Wahid Hasyim	
14/07/2003 1:07:00	19/07/2003 12:58:35	Indonesia Upayakan Tekan Kasus Pelanggaran HAM di NAD	
14/07/2003 0:30:00	19/07/2003 12:58:35	Ketua BEM UI: Korban Tewas Mahasiswa UI Murni Musibah	
14/07/2003 1:55:00	19/07/2003 12:58:34	Jepang Bangun Akademi Politeknik di Sabang	
14/07/2003 2:15:00	19/07/2003 12:58:34	APTR Hanya Bela Kepentingan Pabrik Gula	
14/07/2003 3:18:00	19/07/2003 12:58:34	Darurat Militer Tak Pengaruhi Sektor Wisata di NAD	
14/07/2003 4:05:00	19/07/2003 12:58:33	Ditpostel Tengahi Masalah Telkom-Indosat	
14/07/2003 4:18:00	19/07/2003 12:58:33	Pasukan Nato Tiba di Kabul	
14/07/2003 9:02:00	19/07/2003 12:58:32	Soeyono: Kasus 27 Juli Akan Dipertieskan	
14/07/2003 8:51:00	19/07/2003 12:58:32	Saham Bank Mandiri Mulai Diperdagangkan Hari Ini	
14/07/2003 7:36:00	19/07/2003 12:58:32	Al Qaeda Mengaku Dalangi Serangan di Irak	

Gambar 5.20 Tabel setelah *link-link* dan berita masuk untuk Kompas

Dari gambar 5.20 dan gambar 5.21 dapat disimpulkan bahwa pengambilan informasi berita telah berhasil dilakukan dan dimasukkan ke *database*. Untuk selanjutnya, *client* akan mengakses berita-berita ini secara *offline*, atau langsung mengambilnya dari *database*. Gambar 5.22 untuk kompas dan gambar 5.23 untuk kompas2 menunjukkan client mengakses isi berita tersebut.

NPName	PDName	LiURL	LiContent
Kompas2	Kesehatan	http://www.kompas.co.id/kesehatan/news/0304/18/230607.htm	Dunia dilanda kepanikan luar biasa
Kompas2	Kesehatan	http://www.kompas.co.id/kesehatan/news/0307/11/090805.htm	Kini penderita gangguan hemoroid
Kompas2	Kesehatan	http://www.kompas.co.id/kesehatan/news/0304/18/231005.htm	Makanlah apel setiap hari dan
Kompas2	Kesehatan	http://www.kompas.co.id/kesehatan/news/0304/18/231436.htm	Macam-macam cara orang mengatasi
Kompas2	Kesehatan	http://www.kompas.co.id/kesehatan/news/0307/11/095205.htm	Harian Kompas bekerja sama dengan
Kompas2	Kesehatan	http://www.kompas.co.id/kesehatan/news/0307/11/092010.htm	PEREMPUAN berkacamat adalah
Kompas2	Kesehatan	http://www.kompas.co.id/kesehatan/news/0307/11/091243.htm	SEHARI setelah lahir, Damar Yulianti
Kompas2	Kesehatan	http://www.kompas.co.id/kesehatan/news/0307/11/180409.htm	Bisik-bisik di antara kita soal dugaan
Kompas2	Kesehatan	http://www.kompas.co.id/kesehatan/news/0307/11/082859.htm	Memelihara kesehatan otak bukan

LiContent	WktBerita	WktMasuk	Judul
Dunia dilanda kepanikan luar biasa	11/07/2003 18:03:00	19/07/2003 12:56:22	Aneka Ramuan Pencegah SARS..!
Kini penderita gangguan hemoroid	11/07/2003 18:03:00	19/07/2003 12:56:21	Cara Baru Mengusir Wasir Tanpa Nyeri..!
Makanlah apel setiap hari dan	11/07/2003 18:03:00	19/07/2003 12:56:21	Apel Buah Ajaib Penangkal Penyakit..!
Macam-macam cara orang mengatasi	11/07/2003 18:03:00	19/07/2003 12:56:21	Cabe Anti Mabuk..!
Harian Kompas bekerja sama dengan	11/07/2003 18:03:00	19/07/2003 12:56:20	Konsultasi: Cemburu & Pacar Hobi Selingkuh..!
PEREMPUAN berkacamat adalah	11/07/2003 18:03:00	19/07/2003 12:56:20	Orang Gila di Tengah Stresnya Penduduk Jakarta..!
SEHARI setelah lahir, Damar Yulianti	11/07/2003 18:03:00	19/07/2003 12:56:20	Anak Balita Pun Telantar di Ibu Kota..!
Bisik-bisik di antara kita soal dugaan	11/07/2003 18:03:00	19/07/2003 12:56:19	Betulkah Dengkul Bisa Kopong..?
Memelihara kesehatan otak bukan	11/07/2003 18:03:00	19/07/2003 12:56:19	Chintami Almanagara: Menjaga Posisi Tubuh Tetap Tegak..!

Gambar 5.21 Tabel setelah *link-link* dan berita masuk untuk Kompas2

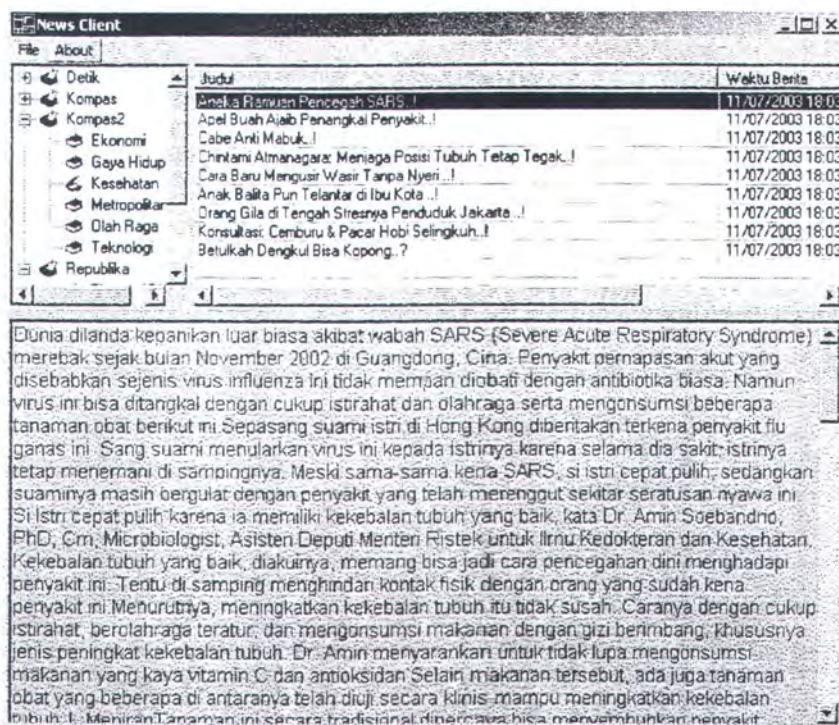
Judul	Waktu Berita	Waktu Masuk
Al Qaeda Mengaku Delangi Serangan di Irak	14/07/2003 7:36:00	18/07/2003 20:00:53
Terseret Sekdu Kamenan Wartawan Kanada	14/07/2003 4:23:00	18/07/2003 20:00:53
Pausulus Nato Tiba di Kabul	14/07/2003 4:18:00	18/07/2003 20:00:54
Ditpostel Tengahai Masalah Telkom-Indosat	14/07/2003 4:05:00	18/07/2003 20:00:54
Darurat Militer Tak Pengaruhi Sektor Wisata di NAD	14/07/2003 3:19:00	18/07/2003 20:00:55
APTR Hanya Bela Keperinginan Pabrik Gula	14/07/2003 2:15:00	18/07/2003 20:00:55
Jepang Bangun Akademi Politeknik di Sabang	14/07/2003 1:55:00	18/07/2003 20:00:55
Indonesia Upayakan Tekan Kasus Pelanggaran HAM di NAD	14/07/2003 1:07:00	18/07/2003 20:00:56
Ketua BEM UI: Korban Tewas Mahasiswa UI Murni Musibah	14/07/2003 0:30:00	18/07/2003 20:00:56
Pesawat Latih TNI AU Jatuh di Waduk Jatiuhur	14/07/2003 16:13:00	18/07/2003 20:00:57
Kapolda Kekuatannya Berbahaya	14/07/2003 11:54:00	18/07/2003 20:00:57
Jenis Ledakan di DPR/MPR Mirip dengan Ledakan di Wahid Hasyim	14/07/2003 11:29:00	18/07/2003 20:00:58
Tiga Direktur PT DI Tolak Kebijakan <i>quot;Merumahkan</i> quot; Karyawan	14/07/2003 10:40:00	18/07/2003 20:00:58
Ribuan Karyawan PT DI Long March ke DPRD Jabar	14/07/2003 9:59:00	18/07/2003 20:00:58
Karyawan Dirgantara Indonesia (PT-DI) yang menyatu dengan kompleks PT-DI	14/07/2003 8:05:00	18/07/2003 20:00:58

Kompas/MH Samsul HadiDirigatja Paskhas TNI AU - Kantor PT Dirgantara Indonesia (Persero) yang berpusat di Bandung, Jawa Barat, dijaga ketat oleh anggota Pasukan Khas (Paskhas). TNI AU menyalurkan keputusan perusahaan itu merumahkan seluruh karyawan PT Dirgantara Indonesia. Pengrumahan selama enam bulan itu dilakukan PT Dirgantara Indonesia karena tidak mampu lagi membiayai operasional perusahaan. Karyawan PT Dirgantara Indonesia (DI) yang semula akan tetap bekerja meski sudah "dirumahkan" oleh perusahaan, tertahan di pintu masuk area kompleks PT DI yang menyatu dengan kompleks TNI AU.

Ketua Umum Serikat Pekerja Forum Komunikasi Karyawan PT-DI (SPFKK-PT-DI) Arief Minardi mengatakan, karyawan yang akan tetap bekerja terpaksa harus dikonsentrasi dahulu di depan patung Husein Sastranegara, Jalan Padjadjaran, Bandung, Senin (14/7).

Menurutnya, sikap demikian tidak terlepas setelah melihat perjagaan cukup ketat dari aparat keamanan yang

Gambar 5.22 Aplikasi client-Melihat Isi Berita kompas



Gambar 5.23 Aplikasi client-Melihat Isi Berita kompas2

5.2.4 Uji Coba *Browsing*

Pada uji coba ini akan dilakukan *browsing* terhadap situs detik dengan direktori *News*, URL-nya adalah:

<http://www1.detik.com/indeksberita/index.cfm?fuseaction=index.detikNews>

Kemudian dipilih salah satu *link* berita, misalnya untuk *link* <http://www.detik.com/peristiwa/2003/07/22/20030722-172057.shtml> yang berjudul “PDIP Calonkan Rosihan Arsyad Sebagai Gubernur Sumsel”. Setelah *main content* dari *link* tersebut terbuka seperti ditunjukkan pada gambar 5.24, halaman yang berisi *content* berita tersebut disimpan ke dalam komputer.

DIGITAL LIFE

detikcom

bii

Informasi dan Reservasi 24 Jam (Pulsa Lotte)

0807-1-GARUDA
4 2 7 8 3 2

Layanan Telepon Penjualan 021-23579999

No. Call Center ini juga dapat diakses melalui pesan (sementara hanya Telkomsel)

Garuda Indonesia

[detikNews](#) [detikFinance](#) [detikFood](#) [detikHot](#) [detiki-Net](#) [detikSport](#) [Foto](#) [Info Iklan](#) [Indeks](#)

[detikHealth](#) [detiksaham](#) [detikTours](#) [SMSiklan](#) [Sepakbola](#) [Ticket Box](#) [Layanan TI](#) [detikMobile](#)

Klik BCA

WLA

Click here for your chance to...

ABILARYA DESAIN INDONESIA 2003

Berita Lain

22/7/2003 16:20 WIB

PDIP Calonkan Rosihan Arsyad Sebagai Gubernur Sumsel

Kontributor : Taufik Wijaya

detikcom - Palembang , PDIP akhirnya memilih Rosihan Arsyad sebagai calon Gubernur Sumatera Selatan (Sumsel) periode 2003-2006. Rosihan Arsyad, yang kemungkinan besar akan berpasangan dengan Radjab Semendawai ini, saat ini adalah Gubernur Sumsel.

"Setahu saya DPP (PDIP) menghendaki Rosihan tampil kembali," kata Bambang Sugianto, salah seorang Fraksi PDIP DPRD Sumsel kepada pers. Selesai (22/7/2003).

Menurut Ketua Komisi A DPRD Sumsel ini, bila ternyata nama yang dikeluarkan DPP PDIP itu tidak direspon anggota FPDI, maka siap-siapa anggota FPDI akan terkena sanksi. "Kalau instruksi pusat tidak didengarkan, berarti orang tersebut sama saja dengan bunuh diri," kata Sekretaris DPW Banteng Muda Indonesia (BMI) Sumsel ini.

Sementara itu terjadi perang klaim antara tim sukses Rosihan Arsyad-Radjab Semendawai dengan pasangan Syahrial-Oesman-Mahyudin.

"Kami optimis pasangan Rosihan-Radjab menang, sebab saat ini kami sudah didukung 47 suara dari lima fraksi," kata Hindarto, Ketua Himpunan Masyarakat Peduli Pembangunan Sumsel, kepada pers, Rabu (22/7/2003).

Menurut dia, pasangan Rosihan-Radjab ini cukup bagus. Rosihan memang bukan kader PDIP. Dia mempunyai later belakang militer. Sedangkan Radjab dari birokrat. "Keduanya cukup berpengalaman," puji Hindarto.

Sementara Johandi Cahya, tim sukses Syahrial-Mahyudin, mengklaim pasangan itu telah didukung suara 45-47 orang. "Pasangan ini cukup unggul. Perpaduan akademisi (Mahyudin) dan birokrat (Syahrial), jelas lebih baik dari pasangan lain," katanya. (aey)

[Print artikel](#) | [Kirim ke teman](#)

SMSiklan

[Lihat](#) [Pasang](#)

Relion
The Most Reliable Computer

Sama-sama berkaki empat!

Siapa yang paling jeli melihat jauh ke depan?

Dengan Relion Anda menggunakan banyak bahan perawatan komputer.

CALL CENTER
0807-1-735466
[www.reliontechnology.com](#)

PAMPAM
[www.pampam.web.id](#)

Tapi ikarun perlu media yang berjuring-juring

detikWEB

Gambar 5.24 Hasil Browsing Situs Detik

Maka hasil yang didapat adalah halaman HTML berserta gambar-gambarnya dan mungkin juga *file-file* lainnya seperti *script* dan sebagainya ikut tersimpan. Setelah disimpan, ukuran *file* yang didapat adalah 94,5 KB sedang *size-on-disk*-nya 184 KB.

Kemudian kita mencoba mengambil berita yang sama dengan menggunakan aplikasi. Dalam hal ini untuk memudahkan uji coba digunakan fungsi *parseDirectory* dari *web service*. Proses ini ditunjukkan pada gambar 5.24.

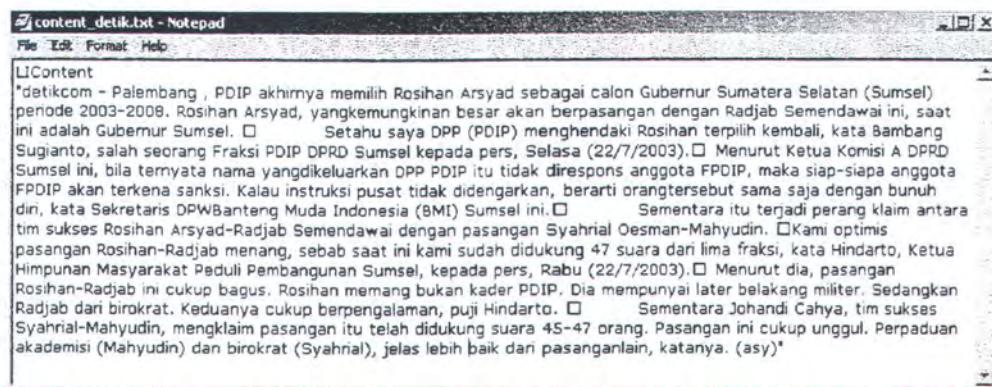
Parameter	Value
NPName:	Detik
PDName:	News
includeContent:	true
Invoke	

Gambar 5.25 ParseDirectory untuk Mengambil Detik News

Setelah *link-link* masuk ke *database* seperti ditunjukkan pada gambar 5.26, dengan membandingkan *link* yang sama seperti pada waktu browsing. Isi/*content* yang ada dalam *database* untuk *link* yang bersangkutan tersebut dibuka dengan *Notepad* seperti ditunjukkan pada gambar 5.27 dan dilihat ukuran *file*-nya yang dalam kasus ini berarti berekstensi *txt*.

NPName	PDName	LIURL	LIContent
Detik	News	http://www.detik.com/peristiwa/2003/07/22/20030722-172057.shtml	detikcom - Palembang , PDIP
Detik	News	http://www.detik.com/peristiwa/2003/07/22/20030722-182002.shtml	detikcom - Jakarta, Gerakan A
WktBentra	WktMasuk	LJudul	
22/07/2003 17:21:00	22/07/2003 18:26:05	PDIP Calonkan Rosihan Arsyad Sebagai Gubernur Sumsel	
22/07/2003 18:20:00	22/07/2003 18:25:38	JPU Ema Diduga Kolusi dengan Pengacara Terdakwa Narkoba	

Gambar 5.26 Link yang Bersangkutan Setelah Masuk ke Database



Gambar 5.27 Isi Link Berita Dibuka dengan Notepad

File ini disimpan dengan nama *content_detik.txt*. Ukuran dari *file* tersebut adalah 1,49 KB dan *size-on-disk*-nya 4,00 KB. Ukuran ini jauh lebih kecil dibandingkan dengan ukuran *file* yang disimpan pada saat *browsing*. Sehingga kesimpulan dari uji coba ini adalah bahwa dengan pengambilan berita melalui aplikasi ini menghasilkan ukuran *file* yang jauh lebih kecil daripada *browsing* secara langsung.



BAB VI

KESIMPULAN DAN SARAN

BAB 6

KESIMPULAN DAN SARAN

Bab ini berisi beberapa kesimpulan dari tugas akhir dan kemungkinan pengembangan lebih lanjut yang dapat dilakukan dari tugas akhir ini

6.1 Kesimpulan

- a) Untuk mengambil informasi berita utama/*main content* dari suatu situs berita yang memiliki pola/struktur situs yang tetap atau memiliki web content management, digunakan suatu formula *Regular Expression* yang sesuai dengan pola situs masing-masing.
- b) Perancangan dan pembuatan skenario *Regular Expression* dapat dilakukan dengan mempelajari struktur/pola suatu situs.
- c) Formula *Regular Expression* dirumuskan dalam satu *file* konfigurasi untuk setiap penyedia berita oleh *Administrator Regex* yang kemudian *file* konfigurasi tersebut akan diimport dan dimasukkan ke database melalui aplikasi *News Administrator* oleh *Administrator User*.
- d) Satu situs penyedia berita bisa memiliki lebih dari satu pola/struktur situs sehingga memiliki lebih dari satu formula *Regular Expression*.
- e) Teknologi *Windows Service* digunakan sebagai aplikasi yang berjalan dalam proses *background* untuk melakukan pengambilan berita secara otomatis.

- f) Teknologi *Web Service* dengan protokol HTTP dapat menjadi penghubung antara *Server* dan *Client*.

6.2 Saran

- a) Penggunaan *pattern recognizion* untuk dapat mengenali pola/struktur situs secara dinamis.
- b) Penelitian lebih lanjut untuk dapat menghasilkan formula *Regular Expression* secara otomatis.

DAFTAR PUSTAKA

DAFTAR PUSTAKA

- [01] Haryanto, Steve. *Majalah Masterweb MWMag*. Indonesia: Oktober 2001, November 2001, issue 04, issue 06
- [02] Jain, Jayesh. *Creating A Windows Service in VB.NET*. www.devarticles.com. Oktober 2002
- [03] JISC TechWatch Report: *Content Management Systems*.
http://www.jisc.ac.uk/techwatch/reports/tsw_01-02.pdf. September 2001
- [04] Microsoft Service Digital Network (MSDN). *.NET Framework Developer's Guide .NET Framework Regular Expressions*,
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconcomregularexpressions.asp>, 2002
- [05] *Regex Library*; www.regexlib.com
- [06] Regular Expresison di .NET;
<http://www.techmedia-online.com/tutorials/t02070101.asp?id=1953>
- [07] Rockford Lhotka. *Implementing a Background Process in Visual Basic .NET*,
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnadvnet/html/vbnet09272002.asp>, Magenic Technologies, 2002
- [08] Thai, Thuan, Huang Q. Lam. *.NET Framework Essential*. 1st Edition, O'Reilly, 2001
- [09] *VB.NET Fast Track*. Syngress, 2001
- [10] Situs-situs berita; www.republika.co.id, www.jawapos.co.id,
www.kompas.co.id, www.surya.co.id, www.detik.com,

www.reuters.com, www.astaga.com, news.bbc.co.uk dan situs-situ berita lainnya

