APLIKASI WEB CRAWLER UNTUK WEB CONTENT PADA MOBILE PHONE

Sarwosri[1] Ahmad Hoirul Basori[1] Wahyu Budi Surastyo[1]

[1]Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Email: sri@its-sby.edu, hoirul@its-sby.edu

ABSTRACT

Crawling is the process behind a search engine, which served through the World Wide Web in a structured and with certain ethics. Applications that run the crawling process is called Web Crawler, also called web spider or web robot. The growth of mobile search services provider, followed by growth of a web crawler that can browse web pages in mobile content type. Crawler Web applications can be accessed by mobile devices and only web pages that type Mobile Content to be explored is the Web Crawler.

Web Crawler duty is to collect a number of Mobile Content. A mobile application functions as a search application that will use the results from the Web Crawler. Crawler Web server consists of the Servlet, Mobile Content Filter and datastore. Servlet is a gateway connection between the client with the server. Datastore is the storage media crawling results. Mobile Content Filter selects a web page, only the appropriate web pages for mobile devices or with mobile content that will be forwarded.

Keywords: *search engine, web crawler, mobile device, mobile content, servlet*

ABSTRAK

Crawling adalah proses di belakang sebuah search engine, yang bertugas menelusuri World Wide Web secara terstruktur dengan etika-etika tertentu. Aplikasi yang menjalankan proses crawling disebut Web Crawler, atau disebut web spider atau web robot. Tumbuhnya penyedia jasa pencarian mobile, menyebabkan tumbuhnya kebutuhan akan web crawler yang dapat menelusuri halaman-halaman web yang bertipe mobile content. Aplikasi Web Crawler dapat diakses oleh peralatan mobile dan hanya halaman-halaman web yang bertipe Mobile Content yang akan ditelusuri Web Crawler ini.

Web Crawler bertugas untuk mengumpulkan sejumlah Mobile Content. Sebuah aplikasi mobile berfungsi sebagai aplikasi pencarian yang akan memanfaatkan hasil dari Web Crawler. Server Web Crawler terdiri dari Servlet, Mobile Content Filter dan Datastore. Servlet merupakan portal koneksi antara client dengan server. Datastore merupakan media-media penyimpanan hasil crawling. Mobile Content Filter menyeleksi suatu web page, hanya web page yang sesuai untuk peralatan mobile atau berisi mobile content yang akan diteruskan.

Kata Kunci: *search engine, web crawler, mobile device, mobile content, servlet*

Tren yang muncul saat ini, pengguna internet lebih sering mengakses internet dari *search engine* daripada langsung ke portal tertentu. Tren ini bukan hanya terjadi pada pengguna internet dari PC tetapi juga terjadi pada pengguna *Mobile* internet [1]. Jika sebelumnya pengguna lebih banyak mengakses ke suatu portal, seperti Yahoo, sekarang pengguna lebih memilih menggunakan *search engine* untuk mengakses web secara lebih luas. Hal tersebut menyebabkan pencarian secara Mobile mendominasi akses informasi bagi penggunanya, seperti halnya pada *World Wide Web*. Dan tentu saja terjadi perkembangan signifikan aktifitas industri *search engine* untuk peralatan *Mobile* dari penyedia jasa *search engine* yang sudah terkenal. Google dan Yahoo telah merilis sejumlah solusi pencarian *Mobile* termasuk pencarian secara lokal dan pencarian melalui SMS. Selain mereka, mulai bermunculan penyedia jasa baru yang menyediakan jasa pencarian secara *Mobile*, seperti Mooobl, 4info, UpSnap, dan Technorati Mobile.

Crawling adalah proses di belakang sebuah *search engine*, yang bertugas menelusuri *World Wide Web* secara terstruktur dengan etika-etika tertentu. Aplikasi yang menjalankan proses crawling disebut *Web Crawler*, atau disebut juga *web spider* atau *web robot*. Web crawler bertugas menelusuri setiap link pada halaman Web di internet dan menyimpannya untuk digunakan lebih lanjut [2]. Tumbuhnya penyedia jasa pencarian mobile, menyebabkan tumbuhnya kebutuhan akan web crawler yang dapat menelusuri halaman-halaman web yang bertipe *mobile content*. *Mobile Content* adalah halaman-halaman web yang dapat ditampilkan oleh peralatan *Mobile*.

WEB CRAWLER

Definisi Web Crawler

Web Crawler adalah sebuah program yang melintasi struktur hypertext dari web, dimulai dari sebuah alamat awal (yang disebut seed) dan secara sekursif mengunjungi alamat web di dalam halaman web. Web Crawler juga dikenal sebagai web robot, spider, worm, walker dan wanderer. Semua search engine besar menggunakan crawler yang mampu melintasi internet secara terus-menerus, untuk menemukan dan mengambil halaman web sebanyak mungkin. Selain untuk search engine, web crawler juga digunakan untuk beberapa penelusuran khusus, seperti implementasi penelusuran alamat email. Hal tersebut mengakibatkan jumlah dan variasi dari web crawler juga semakin banyak.

Usia teknologi Web Crawler bisa dikatakan hampir seumur dengan web [3]. Aplikasi crawler pertama adalah World Wide Web Wanderer. Crawler tersebut merupakan

```
User-agent: *
Disallow: /cgi-bin/
Disallow: /tmp/
Disallow: /~joe/
```

Gambar 1: Format file robots.txt untuk sebuah web server

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">
```

Gambar 2: DocType untuk WML 1.x

kreasi dari Matthew Gray, mahasiswa jurusan Fisika dari Massachusetts Institute of Technology (MIT). World Wide Web Wanderer diciptakan untuk menelusuri pertumbuhan dari World Wide Web yang baru saja lahir. Perkembangan crawler kemudian diikuti dengan proyek Stanford Google, yang menjadi cikal bakal lahirnya search engine Google [1].

Etika Web Crawler

Proses crawling secara terus-menerus dapat menyebabkan beban berlebih pada server suatu web, bukan hanya dari proses penelusuran link tapi juga dari proses pengunduhan halaman web yang bisa berjumlah ratusan. Selain itu pada beberapa web terdapat bagian web yang diharap tidak dimasuki proses crawling. Untuk mengatasi masalah-masalah tersebut, dibuatlah suatu protokol yang disebut Robot Exclusion Protocol [4]. Protokol pertama kali diajukan oleh anggota mailing list robot di tahun 1994 dan kemudian menjadi Internet Draft di tahun 1997. Protokol ini sampai saat ini tidak memiliki standar resmi untuk strukturnya, sehingga protokol ini menjadi semacam aturan tidak tertulis untuk proses crawling.

Robot Exclusion Protocol berbentuk file teks yang berformat robots.txt, yang berisi mekanisme dari suatu web server. Dalam file tersebut dispesifikasikan bagian mana dari suatu web server yang dapat dan yang tidak dapat ditelusuri. Selain itu dalam file itu dapat dispesifikasikan web crawler apa yang dapat menelusuri web server tersebut. Contoh format suatu file robots.txt untuk sebuah web server ditunjukkan pada Gambar 1.

Algoritma Web Crawler

Konsep dari algoritma *Breadth-First Crawler* yang dipakai dalam proses crawling di makalah ini mengecek setiap link dalam suatu halaman web sebelum berpindah ke halaman lain [5]. *Breadth-First Crawler* akan menelusuri setiap link pada halaman pertama, lalu menelusuri setiap link pada halaman dari link pertama di halaman pertama, dan seterusnya. Penelusuran dilakukan sampai tidak ada lagi link baru yang dapat ditelusuri atau jika jumlah halaman yang ditelusuri sudah mencapai batas maksimal yang ditentukan. Algoritma *Depth-First Crawler* memiliki kebalikan proses dari *Breadth-First Crawler*. *Depth-First Crawler* menelusuri semua kemungkinan jalur dari suatu link sampai mencapai suatu dasar sebelum melanjutkan penelusuran ke link berikutnya.

```
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile
1.0 //EN" "http://www.wapforum.org/DTD/
xhtml-mobile10.dtd">
```

Gambar 3: DocType untuk XHTML-MP

```
<!DOCTYPE html PUBLIC"-//WAPFORUM//DTD WML 2.0//EN"
"http://www.wapforum.org/DTD/wml20.dtd">
```

Gambar 4: DocType untuk WML 2.x

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD Compact HTML 1.0
Draft//EN">
```

Gambar 5: DocType untuk C-HTML

```
<meta name="CHTML" content="yes">
```

Gambar 6: Tag Meta untuk C-HTML

```
<meta http-equiv="Content-Type" content="application/
vnd.wap.xhtml+xml">
```

Gambar 7: Tag Meta untuk XHTML-MP

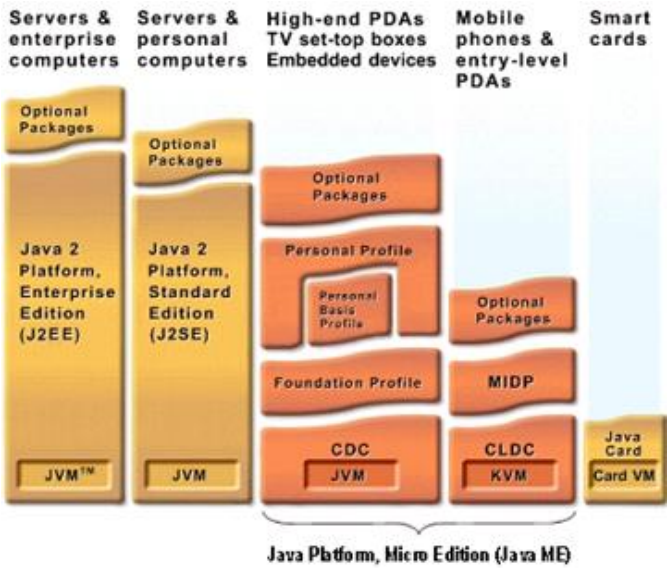
IDENTIFIKASI MOBILE CONTENT

Pendeteksian mobile content dilakukan dalam beberapa langkah yang diawali dengan menggunakan DOCTYPE yang membedakan DTD pada beberapa document XML. DOCTYPE dari masing-masing format mobile content ditunjukkan pada Gambar 2 - Gambar 7. Langkah kedua dilakukan dengan membaca HTTP CONTENT-TYPE pada header respon yang dapat digunakan untuk mengenali tipe data yang diinginkan. Tipe data yang akan didapat pada masing-masing format adalah
- WML untuk "text/vnd.wap.wml",
- XHTML-MP untuk "application/vnd.wap.xhtml+xml",
- dan C-HTML untuk "text/html".

Karena tipe data untuk C-HTML sama dengan HTML pada umumnya, maka cara ini hanya digunakan untuk mengenali WML dan XHTML-MP. Untuk mobile content jenis C-HTML dan XHTML-MP dapat juga dideteksi dari tag meta. Sintaks untuk format C-HTML dan XHTML-MP dapat dilihat pada Gambar 6 dan Gambar 7.

J2ME: Java 2 Micro Edition

Java ME adalah lingkungan pengembangan yang didesain untuk menggunakan aplikasi Java pada peralatan elektronik kecil, seperti telepon seluler, PDA, dan sejenisnya. Untuk mengatasi keterbatasan yang berhubungan dengan pembuatan aplikasi pada peralatan elektronik kecil teknologi Java ME disesuaikan dengan keterbatasan memori, tampilan dan tenaga [6]. Gambar 8 menunjukkan gambaran tentang platform Java. J2ME Device memiliki fitur-fitur yang berbeda. J2ME Configuration ini dirancang untuk menyediakan library standar yang mengimplementasikan fitur standar dari sebuah Handled Device.



Gambar 8: Perbandingan Java ME dengan teknologi Java lainnya

Tabel 1: Perbandingan CLDC dan CDC

CLDC	CDC
Mengimplementasikan subset dari J2SE	Mengimplementasikan seluruh fitur dari J2SE
JVM yang digunakan lebih dikenal dengan KVM	JVM yang digunakan lebih dikenal dengan CVM
Digunakan pada perangkat handled dengan ukuran memori terbatas (160-512 Kbytes)	Digunakan pada perangkat handled dengan ukuran memori minimal 2 Mbytes
Prosesor: 16 Bit atau 32 Bit	Prosesor: 32 Bit

Ada dua macam kategori J2ME saat ini diantaranya adalah CLDC (Connected Limited Device Configuration) dan CDC (Connected Device Configuration). CLDC umumnya digunakan untuk aplikasi Java pada ponsel semacam Nokia, Siemens, PDA, Palm, PocketPC dan two way pagers dengan memori standar 160-512 Kbytes. Sedangkan CDC umumnya digunakan untuk aplikasi Java pada perangkat Handled Device dengan ukuran memori paling tidak 2 MB. Perbandingan keduanya bisa dilihat pada Tabel 1.

METODOLOGI

Pada penelitian ini dibangun suatu server *Web Crawler* yang mengumpulkan sejumlah mobile content web. Selain itu implementasi aplikasi mobile sebagai aplikasi pencarian yang akan memanfaatkan indeks hasil dari *Web Crawler* juga dilakukan. Tahapan yang dikerjakan pada penelitian ini meliputi analisa sistem, perancangan dan implementasi. Sistem dibedakan menjadi sistem pada client dan server.

Analisa Sistem

Sistem Client

Berikut adalah fitur-fitur fungsi yang disediakan oleh sistem di sisi client.
1) melakukan crawling ke server dari file XML.
2) mengubah alamat server yang digunakan.
3) melihat daftar crawling yang telah dilakukan.
4) melihat halaman web hasil crawling dalam browser.
5) mendaftarkan client dengan unique id.
6) meminta server untuk memulai proses crawling.
7) mengecek status proses crawling.
8) meminta hasil proses crawing.
9) meminta halaman web dari suatu URL hasil crawling.

Sistem Server

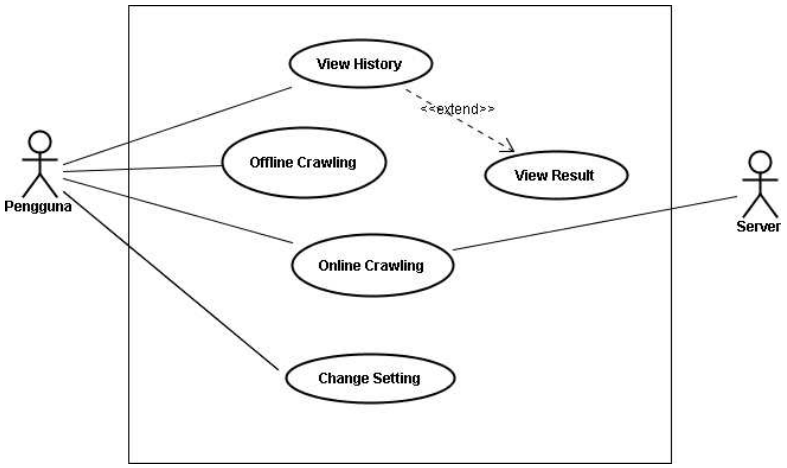
Berikut adalah fitur-fitur fungsi yang disediakan oleh sistem di sisi server. Server dapat melakukan crawling dari file XML, jika timestamp file XML kurang dari 6 jam. Server juga dapat melakukan crawling dari database, jika timestamp file XML lebih atau sama dengan 6 jam. Selain itu server masih tetap dapat melakukan crawling secara online jika tidak ditemukan data crawling di file XML dan di database. Terkait dengan hasil crawling, server dapat menyimpan hasil tersebut ke file XML maupun ke database.

Pada Gambar 9 ditunjukkan diagram use case untuk client, sedangkan pada Gambar 10 diberikan diagram use case untuk server.

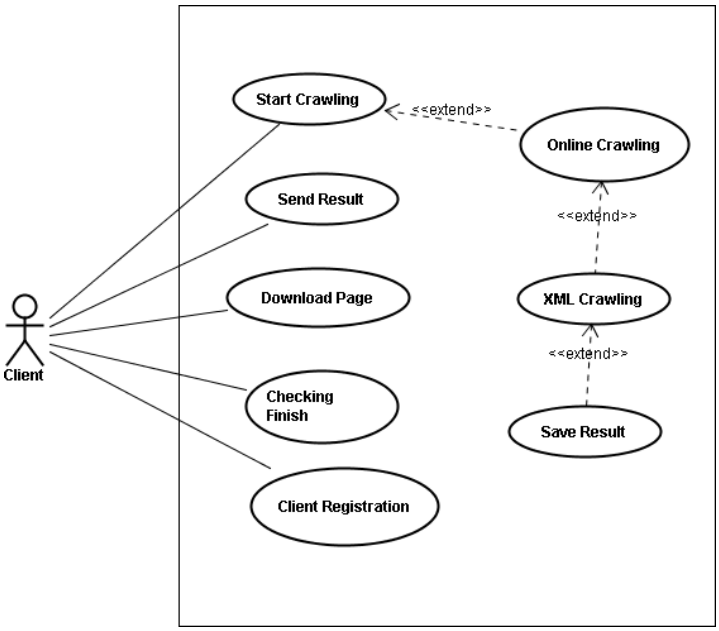
Perancangan

Pada bagian perancangan data diberikan dua gambaran pemodelan. Rancangan pertama adalah rancangan model data konseptual (*Conceptual Data Model*, CDM). Sedangkan rancangan yang kedua adalah rancangan model data fisik (*Physical Data Model*, PDM). Pada Gambar 11 ditunjukkan rancangan data fisik PDM yang akan menjadi tabel di database.

Perangkat lunak untuk integrasi kebutuhan non fungsional pada diagram use case dan skenario ini menggu-



Gambar 9: Diagram use case Client



Gambar 10: Diagram use case Server

nakan konsep hirarki. Jadi keseluruhan aplikasi pada form-form yang ada akan menjadi satu kesatuan dalam form utama. Struktur menu yang muncul pada form utama ditunjukkan pada Gambar 12.

Implementasi

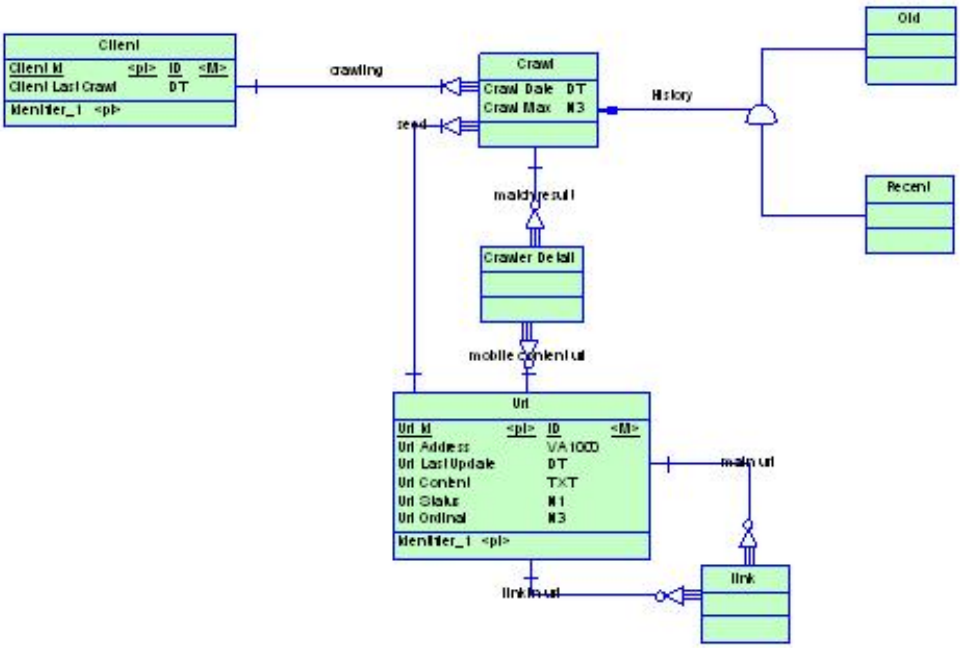
Untuk lingkungan implementasi membutuhkan spesifikasi perangkat keras dan perangkat lunak. Detil informasi tentang spesifikasi dalam pembangunan aplikasi *Instant Message* pada handphone dengan teknologi GPRS dan bluetooth ditunjukkan dalam Tabel 2.

Proses yang telah dirancang menjadi suatu *class diagram* akan diimplementasikan menjadi kelas midlet pada Java ME untuk client dan menjadi servlet pada J2EE untuk server. Implementasi proses dari kelas *Crawler Client* ditunjukkan pada Gambar 13.

Tabel 2: Lingkungan implementasi

Perangkat Keras	Tipe handphone: Sony Ericsson W830 Support: GPRS
Perangkat Lunak	Sistem Operasi: Windows XP (Server) Compiler dan Tools: J2EE 1.4 (Server), WTK 2.5.1 (Client), Netbeans 5.5.1 Library: J2ME-Polish

Rancangan antar-muka yang telah dibuat sebelumnya akan diimplementasikan pada mobile phone. Di dalam an-



Gambar 11: PDM

Gambar 12: Struktur Menu

tarmuka ini terbagi menjadi beberapa halaman, yaitu menu utama, inputform, process, hasil crawling, history, setting-form, dan about. Implementasi antarmuka ditunjukkan pada Gambar 14.

UJI COBA

Parameter keberhasilan uji coba adalah semua fungsi use case pada sisi client dan sisi server bisa dilakukan.

Uji Coba Online Crawling

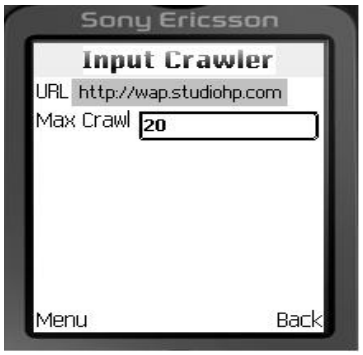
Pada uji coba ini dilakukan dengan cara pengguna handphone memasukkan seed_url dan maksimum crawling lalu memilih online crawling. Adapun langkah-langkah terlihat pada Gambar 15 - Gambar 20.

Pada Gambar 16 pertama-tama pengguna memilih menu [New Crawling]. Kemudian pengguna akan memasukkan seed_url dan maksimum crawling seperti pada Gambar 16. Lalu pengguna mengirim request tersebut dengan memilih [Menu > Send] seperti pada Gambar 17. Setelah itu akan tampil pilihan untuk melakukan online crawling atau offline crawling seperti pada Gambar 18. Pengguna memilih [Ok]. Kemudian proses crawling akan berjalan dan akan muncul tampilan progress dari proses seperti pada Gambar 19. Angka di samping progress itu adalah jumlah URL yang sudah diproses. Setelah proses crawling selesai akan muncul daftar URL hasil crawling seperti pada Gambar 20. Setelah itu pengguna dapat memilih salah satu URL untuk ditampilkan pada browser.

```
public class CrawlerClient extends MIDlet implements CommandListener, Runnable {
    public void displayHasil() {
        String url = hasilList.getString(hasilList.getSelectedIndex());
        Save saveId = new Save("id.db");
        saveId.open();
        String mobileId = saveId.readRecord(1);
        try {
            HttpConnection con = ConnectionUtil.connect(SettingForm.getServerUrl());
            String input = "@sendHtml "+url;
            ConnectionUtil.sendInput(con,mobileId,input);
            String html = ConnectionUtil.getData(con);
            browser.loadPage(html);
            display.setCurrent(browser);
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}
```

Gambar 13: Kelas Crawler Client.

Gambar 14: Halaman Menu Utama



Gambar 16: Input Seed URL dan Max Crawl

Gambar 15: Memilih New Crawling

Gambar 17: Mengirim Request

Uji Coba Offline Crawling

Pada uji coba ini dilakukan dengan cara pengguna handphone memasukkan seed_url dan maksimum crawling lalu memilih offline crawling. Adapun langkah-langkah terlihat pada Gambar 21 - Gambar 23.

Pertama-tama pengguna memilih menu New Crawling seperti pada Gambar 15. Kemudian pengguna akan memasukkan seed_url dan maksimum crawling seperti pada Gambar 16. Lalu mengirim request tersebut dengan memilih menu Menu > Send, seperti pada Gambar 16. Setelah itu, akan tampil pilihan untuk melakukan online crawling atau offline crawling seperti pada Gambar 17. Pengguna bisa memilih Cancel.

Lalu proses crawling akan berjalan, dan akan muncul tampilan progress dari proses seperti pada Gambar 18. Proses tersebut akan berjalan lebih cepat karena proses berlangsung secara offline. Angka di samping progress itu adalah jumlah url yang sudah diproses. Setelah proses crawling selesai, akan muncul daftar url hasil crawling seperti Gambar 19.

Setelah itu pengguna dapat memilih salah satu url untuk ditampilkan pada browser.

Uji Coba View History

Pada uji coba ini dilakukan dengan cara pengguna handphone memilih menu history, seperti pada Gambar 20 dan

Gambar 18: Pilihan Metode Crawling

Gambar 19: Crawling Sedang Diproses

Gambar 20: Daftar URL Hasil Crawling

Gambar 21: Daftar History Crawling

Gambar 22: Menghapus History

Gambar 23: Tampilan Halaman Web Pada Browser

Gambar 21. Setelah itu akan muncul daftar proses crawling yang telah dilakukan, seperti pada Gambar 22. Setelah itu pengguna dapat memilih salah satu dari daftar tersebut unuk ditampilkan detail hasilnya. Maka akan muncul daftar hasil crawling seperti pada Gambar 21. Pengguna juga dapat menghapus daftar tersebut, dengan memilih menu Menu > Delete, seperti pada Gambar 23.

Uji Coba View Result

Pada uji coba ini dilakukan dengan cara pengguna handphone memilih url yang akan ditampilkan pada browser, seperti pada Gambar 20. Setelah itu akan muncul browser, seperti pada Gambar 23.

SIMPULAN

Setelah dilakukan serangkaian uji coba dan analisa terhadap perangkat lunak yang dibuat, maka dapat diambil kesimpulan sebagai berikut. Proses pendahuluan dari crawling yaitu proses koneksi antara client dan server sudah berhasil dilaksanakan. Setelah dilakukan crawling maka proses pengiriman file XML hasil crawling juga sudah berhasil. Untuk kemudahan pengguna melihat hasil crawling, fitur proses filtering Mobile Content dengan menggunakan beberapa identifikator sudah berhasil. Kemudian tampilan akhir pada browser juga sudah berhasil terintegrasi ke aplikasi meskipun tampilan tersebut belum bisa dikatakan sempurna

DAFTAR PUSTAKA

[1] LeClaire, J.: *Mobile Browsing Heads Toward the Mainstream*. (2008)

[2] Takeno, H.: *Developing Web Crawler for Massive Mobile Search Service*. IEEE (2006)

[3] Salahuddin, M.: *Pemrograman J2ME: Belajar Cepat Pemrograman Perangkat Lunak Mobile*. Informatika (2006)

[4] Koster, M.: *A Standard for Robot Exclusion*. (2008)

[5] Schildt Herbert, H.J.: *Crawling The Web With Java*. McGraw-Hill (2005)

[6] Paul J. Timmins, Sean McCormick, E.A.C.E.W.: *Characteristics of Mobile Web Content*. Worcester Polytechnic Institute (2006)