

SKRIPSI

DESAIN SISTEM SINDIKASI WEB MELALUI APLIKASI MOBILE MENGGUNAKAN JAVA (J2ME)



**Disusun Oleh :
IWAN HADI MULYONO
04.12.687**

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER & INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2009**

LEMBAR PERSETUJUAN

**DESAIN SISTEM SINDIKASI WEB MELALUI APLIKASI
MOBILE MENGGUNAKAN JAVA (J2ME)**

SKRIPSI

*Disusun dan Diajukan sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik Komputer dan Informatika Strata Satu (S-1)*

Disusun Oleh :

IWAN HADI MULYO NO


NIM : 04.12.687

Diperiksa dan Disetujui

Dosen Pembimbing I

Dosen Pembimbing II


I Komang Somawirata, ST, MT
NIP.Y. 1030100361


M. Ashar, ST, MT
NIP : Y.103050048

Mengetahui
Ketua Jurusan Teknik Elektro S-1


Ir. F. Yudi Limpraptono, MT
NIP Y. 1039500274

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2009**

KATA PENGANTAR



Puji syukur kehadiran Allah SWT atas segala limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan Skripsi ini dengan baik dan lancar yang berjudul :

"DESAIN SISTEM SINDIKASI WEB MELALUI APLIKASI MOBILE MENGGUNAKAN JAVA (J2ME)"

Skripsi ini disusun sebagai salah satu persyaratan untuk meraih gelar Sarjana Teknik pada Institut Teknologi Nasional Malang.

Penyelesaian laporan skripsi ini tidak lepas dari bantuan banyak pihak yang benar-benar memberikan masukan dan dukungan kepada penulis. Untuk itu penulis mengucapkan banyak terimakasih kepada :

1. **Bapak Ir. F. Yudi Limpraptono, MT**, selaku Ketua Jurusan Teknik Elektro S-1.
2. **Bapak I Komang Somawirata, ST, MT**, selaku Dosen Pembimbing I yang telah meluangkan waktu untuk memberikan bimbingan dan petunjuk selama pelaksanaan tugas akhir.
3. **Bapak M. Ashar, ST, MT**, selaku dosen pembimbing II yang telah mendukung memberikan kepercayaan penuh kepada penulis dalam menyelesaikan tugas akhir ini.
4. **Bapak, Ibu, Kakak-Kakakku, dan Adek - Adekku** (M'Leni, Mz Yayan, Adek Anggra, dan Adek Diva) tercinta, terima kasih atas semua cinta, kasih

sayang, doa dan dukungan yang terus-menerus mengalir. Semoga aku selalu menjadi anak yang shaleh dan berbakti untuk keluarga kecil kita ini.

5. **Bunda_ku (Jibrut)** yang senantiasa mensupport & mendoakan ai2, anugerah terindah yang pernah ai2 miliki yaitu bunda (Mengerti yang Tidak Dijelaskan, Mendengar yang Tidak Dikatakan)
6. **Partnerku Ruhe** yang banyak berperan dalam perancangan tugas akhir ini, semangat ya buat usaha barunya. Aku doakan jadi pengusaha yang sukses. Amin.....!!!!!!!!!!!!!!!
7. **Abi, Umi, Kakak, Adek** yang banyak memberikan nasehat dalam memapaki kehidupan ini menjadi lebih baik.
8. **Nasgor Arudam** yang selalu membantuku jikalau aku membutuhkan kalian,ha..ha..ha..ha...semoga silahturahmi tetap terjalin meskipun kita jauh nanti.
9. **Teman – teman GS** (kempet, naning, cimot, sules, ardik, luhur, kepek, davit) banyak hal yang telah kita lalui bersama.
10. **Teman-teman penghuni apartement simpang candi panggung 63** (Adi, Aris, Hario), meskipun kita seataap belum sampai satu tahun, tapi kita uda seperti saudara. Tetep jaga persaudaraan ini ya....!!!o ya satu lagi penghuni SCP 63 yaitu panda yang telah membuatku g bisa tidur.Bagosssss...!!!!
11. **Mbh Wito** yang senantiasa mendoakan dan membimbingku selama ini.
12. **Pak Jayeng, Pak Sugeng, Bu Puji** yang telah banyak membantu dalam segala urusan persyatan yang berhubungan dengan tugas akhir ini.
13. **Serta semua pihak** yang telah membantu terselesaikannya skripsi ini yang tidak dapat disebutkan satu-persatu

Penulis menyadari bahwa laporan ini masih jauh dari kesempurnaan, baik dari materi maupun teknik penyajiannya. Untuk itu segala kritik dan saran membangun, sangat penulis harapkan. Semoga laporan ini dapat memberikan guna dan manfaatnya.

Malang, Agustus 2009

Penulis

DAFTAR ISI

	Halaman
ABSTRAK	i
KATA PENGANTAR	ii
DAFTAR ISI	iii
DAFTAR TABEL	viii
DAFTAR GAMBAR	ix
BAB I : PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan.....	3
1.4 Batasan Masalah	3
1.5 Metodologi	3
1.6 Sistematika Penulisan.....	4
BAB II : TINJAUAN PUSTAKA	5
2.1 Pengertian Word Wide Web	5
2.2 Pengertian Java.....	7
2.2.1 Pengertian J2ME	8
2.2.1.1 Configuration.....	10
2.2.1.2 Profil	10
2.2.1.3 CLDC	10
2.2.1.4 GCF	11

	2.2.1.5 MIDP	12
	2.2.1.6 Midlet	13
	2.2.1.7 Siklus Midlet	14
	2.3 Web Server Apache	14
	2.4 PHP.....	16
	2.5 WAP	17
	2.6 GPRS	19
BAB III	: PERANCANGAN DAN PEMBUATAN SISTEM	23
	3.1 Spesifikasi Sistem	23
	3.1.1 Spesifikasi Software.....	24
	3.1.2 Spesifikasi Hardware.....	26
	3.2 Desain Keseluruhan Sistem.....	27
	3.2.1 Desain Komunikasi HTTP	28
	3.2.2 Desain Web Sindikasi	34
	3.2.3 Desain Konten	38
BAB IV	: IMPLEMENTASI DAN PENGUJIAN HASIL.....	41
	4.1 Pengujian Koneksi HTTP	41
	4.2 Pengujian Login Pada Web Site Yang Dituju.....	44
	4.2.1 Analisa Pada Web Site www.bola.com.....	45
	4.2.2 Analisa Pada Web Site www.chip.co.id.....	47
	4.2.3 Analisa Pada Web Site www.vivanews.com.....	49
	4.2.4 Analisa Pada Web Site www.Antarane.ws.com	51
	4.3 Analisa Pengujian Update	53

4.4	Analisa Pengujian Kecepatan Koneksi.....	54
4.5	Analisa Pengujian Aplikasi Web Sindikasi Pada Beberapa Jenis HandPhone	55
4.5.1	Tampilan Pada HandPhone Nokia	55
4.5.2	Tampilan Pada HandPhone LG.....	56
4.5.3	Tampilan Pada HandPhone Samsung.....	56
4.5.4	Tampilan Pada HandPhone Siemens.....	57
4.5.6	Tampilan Pada HandPhone Sony Ericsson	57
BAB V	: PENUTUP	58
5.1	Kesimpulan	58
5.2	Saran	59
DAFTAR PUSTAKA	61
LAMPIRAN		

DAFTAR TABEL

	Halaman
Tabel 2.1 Platform Java.....	8
Tabel 3.1 Response Code Http.....	32
Tabel 4.1 Pengujian Kecepatan Koneksi.....	54

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Platform Java.....	8
Gambar 2.2 Arsitektur J2ME.....	9
Gambar 2.3 Hirarki Koneksi GCF.....	12
Gambar 2.4 Siklus Hidup Midlet.....	14
Gambar 2.5 Layer WAP.....	17
Gambar 2.6 Cara Kerja Wap.....	18
Gambar 2.7 Cara Kerja GPRS.....	21
Gambar 3.1 Nokia N73.....	26
Gambar 3.2 Nokia N73.....	27
Gambar 3.3 Desain Keseluruhan Sistem.....	28
Gambar 3.4 Analisa Komunikasi HTTP Secara Hardware.....	29
Gambar 3.5 Flowchart Komunikasi HTTP.....	31
Gambar 3.6 Request RSS Feed.....	34
Gambar 3.7 Flowchart Web Sindikasi.....	36
Gambar 3.8 Desain Konten Web Sindikasi.....	39
Gambar 4.1 Arsitektur Low-End Consumer Device.....	41
Gambar 4.2 Hirarki GCF.....	41
Gambar 4.3 Layer WAP.....	42
Gambar 4.4 Hasil Koneksi HTTP.....	44
Gambar 4.5 Login Dan Hasil Login.....	44

Gambar 4.6	Tampilan Web Site Pada Komputer Dekstop.....	45
Gambar 4.7	Tampilan Pada Aplikasi	45
Gambar 4.8	Tampilan Web Site Pada Komputer Dekstop.....	46
Gambar 4.9	Tampilan Pada Aplikasi	46
Gambar 4.10	Tampilan Web Site Pada Komputer Dekstop.....	47
Gambar 4.11	Tampilan Pada Aplikasi	47
Gambar 4.12	Tampilan Web Site Pada Komputer Dekstop.....	48
Gambar 4.13	Tampilan Pada Aplikasi	48
Gambar 4.14	Tampilan Web Site Pada Komputer Dekstop.....	49
Gambar 4.15	Tampilan Pada Aplikasi	49
Gambar 4.16	Tampilan Web Site Pada Komputer Dekstop.....	50
Gambar 4.17	Tampilan Pada Aplikasi	50
Gambar 4.18	Tampilan Web Site Pada Komputer Dekstop.....	51
Gambar 4.19	Tampilan Pada Aplikasi	51
Gambar 4.20	Tampilan Web Site Pada Komputer Dekstop.....	52
Gambar 4.21	Tampilan Pada Aplikasi	52
Gambar 4.22	Tampilan Sebelum, Sesaat & Sesuda Update	53
Gambar 4.23	Tampilan Pengujian Aplikasi Pada Nokia.....	55
Gambar 4.24	Tampilan Pengujian Aplikasi Pada LG	56
Gambar 4.25	Tampilan Pengujian Aplikasi Pada Samsung.....	56
Gambar 4.26	Tampilan Pengujian Aplikasi Pada Siemens.....	57
Gambar 4.27	Tampilan Pengujian Aplikasi Pada Sony Ericsson	57

BAB I PENDAHULUAN

1. Pendahuluan

Seiring berjalannya waktu dan perkembangan teknologi informasi yang sangat pesat, penggunaan teknologi computer dan internet sangatlah berperan aktif dalam kehidupan kita, khususnya pada bidang pencarian berita terbaru. Kebutuhan akan informasi tersebut akan berkembang pesat seiring dengan tuntutan era yang juga semakin berkembang. Semakin hari kemajuan teknologi komputer, baik itu di bidang piranti perangkat lunak maupun perangkat keras berkebang dengan sangat pesat . di sisi lain juga berkembang kearah yang sangat mudah dari segi pengamplikasian dan murah dalam segi biaya. Solusi untuk bidang dalam segi kerja apapun akan ada cara untuk dapat dilakukan melalau media komputer dan internet termasuk dalam bidang pencarian berita terbaru. Sehingga memudahkan penguna internet atau penjelajahan web site untuk mendapatkan informasi yang uptodate.

1.1 Latar Belakang

Dunia website saat ini sangat berkembang sangat pesat mengamplikasikan teknologi demi terciptanya system yang kuat. Dulu ketika sebuah situs web kebanyakan masih statis, kita menggunakan fasilitas *bookmark* untuk menyimpan informasi tentang sebuah alamat tertentu di sebuah situs web internet. Pengguna Internet Explorer biasa melakukannya dengan menyimpan alamat itu di bagian *favourites*. Tapi di masa sekarang, ketika aktifitas memperbaharui sebuah situs web menjadi semakin sering (misalnya blog, situs berita atau situs lainnya), maka tentu akan perlu upaya ekstra untuk secara rutin menyambangi setiap situs web untuk mencari informasi terbaru. Dengan memanfaatkan teknologi web sindikasi yang menggunakan metode RSS, bukan anda atau audience anda yang mendatangi informasi, melainkan informasi itu yang mendatangi anda atau audience anda. Karena itu RSS sering juga disebut sebagai sebuah metode berlangganan konten internet.

Sindikasi web adalah suatu bentuk sindikasi menyediakan bagian dari suatu situs web untuk digunakan oleh situs lainnya. Ini dapat dilakukan secara sederhana dengan melisiansikan isinya untuk dapat digunakan orang lain. Umumnya, sindikasi web merujuk pada penyediaan umpan web dari suatu situs kepada orang lain untuk

memberikan ringkasan isi yang baru ditambahkan pada situs tersebut, misalnya berita atau kiriman forum terbaru.

RSS adalah singkatan dari *Really Simple Syndication*. Ia memungkinkan sebuah situs web untuk secara rutin mengirim informasi kepada para pengakses atau *to syndicate*. Biasanya ia bekerja dengan cara mengirimkan informasi tentang alamat atau lokasi dimana informasi itu disimpan secara otomatis begitu informasi itu diperbaharui. Lebih dari sekedar alamat, RSS juga memungkinkan isi informasi itu sendiri dikirimkan langsung baik ringkasan ataupun keseluruhan, termasuk pula file-file tertentu, seperti file audio. RSS (atau biasa disebut RSS Feeds) sebenarnya adalah sebuah file yang berisikan informasi terbaru dari sebuah situs web. Untuk melihatnya diperlukan sebuah pembaca RSS atau *RSS Reader* atau juga sering disebut sebagai *RSS Aggregator*. Aggregator itu bisa berbasis web (online) atau berbasis software (offline).

Java adalah sebuah bahasa pemrograman berorientasi pada objek yang sangat fleksibel, dimana kita bisa menjalankan satu program java di berbagai macam platform yang berbeda, java sendiri saat ini terbagi dalam 3 kategori besar, yaitu J2SE (Java 2 Standart Edition) J2EE (Java 2 Enterprise Edition) dan J2ME (Java 2 Micro Edition). Kategori-kategori tersebut memiliki fungsi yang berbeda-beda, yaitu J2SE biasanya digunakan untuk Aplikasi berbasis Desktop, J2EE biasanya digunakan untuk aplikasi berbasis Enterprise (Client - Server), dan J2ME biasanya digunakan untuk aplikasi pada mobile device, sebagai turunan dari J2SE dan merupakan yang paling berkembang pesat saat ini.

Teknologi J2ME saat ini banyak di manfaatkan untuk membangun aplikasi pada mobile device, karena dukungan dari vendor handphone yang kuat menyebabkan perkembangan teknologi inipun semakin pesat, dengan menggunakan teknologi J2ME. Adapun keunggulannya antara lain memudahkan penjelajah Web Site untuk mendapatkan informasi atau berita terbaru dari suatu web site yang dituju menggunakan telepon selular.

Dengan Teknologi J2ME dapat di desain sebuah aplikasi berbasis mobile dimana aplikasi tersebut di desain untuk mempermudah user melakukan koneksi dengan internet untuk mengakses suatu web site, sehingga user bisa mengakses suatu web site dengan lebih mudah, menggunakan telepon selular.

1.2 Rumusan Masalah

Berdasarkan hal di atas maka timbul suatu perumusan permasalahan yang peneliti ajukan yaitu :

Bagaimana menciptakan suatu aplikasi Sistem Sindikasi Web melalui mobile menggunakan JAVA (J2ME) yang mendukung dan mempermudah penjelajah Web Site untuk mendapatkan informasi terbaru saja.

1.3 Tujuan

Adapun tujuan yang ingin dicapai dalam pelaksanaan skripsi ini adalah :

Menghasilkan aplikasi Sistem Sindikasi Web melalui mobile yang mempermudah penjelajah Web Site untuk selalu mendapatkan informasi yang uptodate.

1.4 Batasan Masalah.

Agar permasalahan mengarah sesuai dengan tujuan maka pembahasan dibatasi pada hal-hal sebagai berikut :

1. Dalam merancang sistem hanya pada Sistem aplikasi mobile.
2. Tidak Membahas Teknologi GPRS secara luas, tetapi hanya pada handphone.
3. Aplikasi mobile ini dibatasi hanya pada aplikasi berbasis web.
4. Tidak membahas masalah sistem jaringan yang digunakan.

1.5 Metode Penelitian

Metode yang digunakan dalam penyusunan skripsi ini adalah :

1. Studi literatur, yaitu tinjauan pustaka untuk mempelajari teori-teori yang terkait dan berhubungan dengan permasalahan melalui media Referensi Buku, dan Literatur dari Internet.
2. Perancangan
Merancang dan mengembangkan desain sindikasi web yang sesuai dengan web site yang sudah ada.
3. Pembuatan Program
Pembuatan Aplikasi sindikasi web dengan Bahasa Pemrograman JAVA.
4. Implementasi & Pengujian

Merancang dan mengembangkan desain sindikasi web berdasarkan data-data yang diperoleh.

1.7 Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan skripsi ini, maka peneliti menyajikan secara sistematis sebagai berikut :

Bab I : Pendahuluan

Berisi Latar Belakang, Perumusan Masalah, Tujuan dan Manfaat Penelitian, Pembatasan Permasalahan, Metodologi Penelitian dan Sistematika Penulisan.

Bab II : Tinjauan Pustaka

Berisi tentang landasan teori mengenai permasalahan yang berhubungan dengan pembahasan yang dilakukan, yang di dalamnya memuat teori-teori tentang Web Sindikasi dan JAVA.

Bab III : Desain & Analisis System

Bab ini membahas Desain dan analisis dari Aplikasi Web Sindikasi pada hardware dan software.

Bab IV : Implementasi dan Pengujian Sistem

Berisi tentang Implementasi dari Hasil Desain Aplikasi pada emulator dan pengujian pada aplikasi.

Bab V : Kesimpulan dan saran

Berisi tentang kesimpulan yang dapat diambil berdasarkan hasil uraian pada bab-bab sebelumnya dan saran mengenai hasil yang telah diPeroleh.

BAB II

LANDASAN TEORI

Dalam bab ini akan dibahas mengenai dasar-dasar teori yang mendukung dalam pembuatan Aplikasi Web Sindikasi menggunakan JAVA.

2.1. Word Wide Web

Internet merupakan jaringan global yang menghubungkan suatu network dengan network lainnya di seluruh dunia melalui TCP/IP menjadi protokol penghubung antara jaringan-jaringan yang beragam di seluruh dunia untuk dapat berkomunikasi, tukar menukar berbagai macam jenis informasi, dan data tanpa batas Negara dan Benua. World wide web mendapat perhatian publik yang sangat besar yang tidak dapat disamai oleh aplikasi internet lainnya. Pada tahun 1995, www menggantikan FTP sebagai aplikasi internet yang bertanggungjawab atas sebagian besar lalu lintas internet. Web telah menjadi sedemikian terkenalnya sehingga kadang dicampuradukkan dengan istilah internet itu sendiri meskipun pengertian "di Web" dan "di Inetrnet" sebenarnya tidaklah sama.

Website atau situs dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (hyperlink). Bersifat statis apabila isi informasi website tetap, jarang berubah, dan isi informasinya searah hanya dari pemilik website. Bersifat dinamis apabila isi

informasi website selalu berubah-ubah, dan isi informasinya interaktif dua arah berasal dari pemilik serta pengguna website. Contoh website statis adalah berisi profil perusahaan, sedangkan website dinamis adalah seperti Friendster, Multiply, Okezone, Chip, Detik dll. Dalam sisi pengembangannya, website statis hanya bisa diupdate oleh pemiliknya saja, sedangkan website dinamis bisa diupdate oleh pengguna maupun pemilik.

Untuk mendukung kelanjutan dari situs diperlukan pemeliharaan setiap waktu sesuai yang diinginkan seperti penambahan informasi, berita, artikel, link, gambar atau lain sebagainya. Tanpa pemeliharaan yang baik situs akan terkesan membosankan atau monoton juga akan segera ditinggal pengunjung.

Pemeliharaan situs dapat dilakukan per periode tertentu seperti tiap hari, tiap minggu atau tiap bulan sekali secara rutin atau secara periodik saja tergantung kebutuhan (tidak rutin). Pemeliharaan rutin biasanya dipakai oleh situs-situs berita, penyedia artikel, organisasi atau lembaga pemerintah. Sedangkan pemeliharaan periodik biasanya untuk situs-situs pribadi, penjualan/e-commerce, dan lain sebagainya (www.wikipedia.com).

Pada era sekarang website yang sangat ramai dikunjungi adalah website yang memberikan berita yang update atau disebut juga dengan news website, pada awalnya news website hanya menyampaikan berita secara online. Akan tetapi pada saat ini news website sudah dipenuhi oleh :

1. **Quick Count** : perhitungan cepat yang dilakukan oleh suatu lembaga survei, biasanya digunakan untuk pemilu legislatif dan presiden.
2. **Iklan** : mempromosikan product – product dari suatu perusahaan, dan mempromosikan layanan masyarakat pada department pemerintahan
3. **Jajak Pendapat** : dimana para pengunjung website dapat saring dengan pengguna atau pemilik website dalam segala topic permasalahan

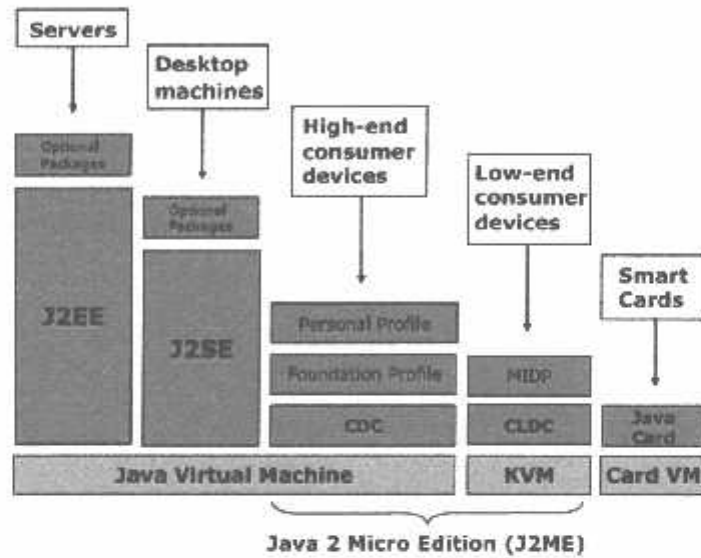
2.2. JAVA

Java dibuat pada tahun 1991 oleh James Gosling. Pada awalnya diberi nama Oak, dimana untuk menghormati pohon yang ada di luar jendela Gosling. Kemudian namanya diubah ke Java karena telah ada sebuah bahasa yang diberi nama Oak. Motivasi sesungguhnya dari Java adalah kebutuhan akan sebuah bahasa yang bisa digunakan pada berbagai platform yang bisa dimasukkan ke dalam berbagai produk elektronik seperti pemanggang roti dan lemari es. Salah satu dari proyek pertama yang dikembangkan menggunakan JAVA sebuah remote kontrol yang diberi nama Star 7.

Pada saat yang sama, World Wide Web dan Internet berkembang sangat cepat. Gosling menyadari bahwa Java dapat digunakan untuk pemrograman Internet. Dengan keluarnya versi 1.2, platform Java telah dipilah-pilah menjadi beberapa edisi : The Standard Edition(J2SE), Enterprise Edition(J2EE), Mobile Edition(J2ME), dan JavaCard API ([www. JENI.co.id](http://www.JENI.co.id)).

Tabel 2.1. Platform Java

J2SE – Java 2 Platform, Standard Edition	Aplikasi Desktop
J2EE – Java 2 Platform, Enterprise Edition	Aplikasi enterprise dengan fokus pada pengembangan sisi webserver, termasuk servlet, JSP, EJB, dan XML
J2ME – Java 2 Platform, Micro Edition	Perangkat Mobile
JavaCard	Smart Cards



Gambar 2.1. Platform JAVA

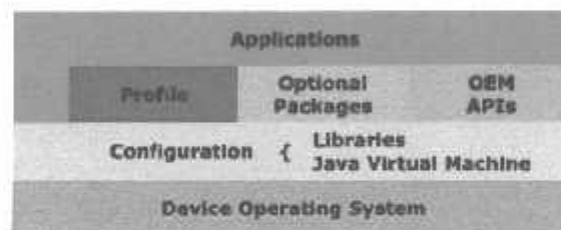
2.2.1. J2ME

J2ME adalah satu set spesifikasi dan teknologi yang fokus kepada perangkat konsumen. Perangkat ini memiliki jumlah memori yang terbatas, menghabiskan sedikit daya dari baterai, layar yang kecil dan bandwidth jaringan yang rendah.

Dengan berkembangnya perangkat mobile konsumen dari telepon, PDA, kotak permainan ke peralatan-peralatan rumah, Java menyediakan suatu lingkungan yang portable untuk mengembangkan dan menjalankan aplikasi pada

perangkat ini. Program J2ME, seperti semua program JAVA adalah diterjemahkan oleh VM. Program-program tersebut dikompil ke dalam bytecode dan diterjemahkan dengan Java Virtual Machine(JVM).Ini berarti bahwa program-program tersebut tidak berhubungan langsung dengan perangkat.

J2ME menyediakan suatu interface yang sesuai dengan perangkat. Aplikasi-aplikasi tersebut tidak harus dikompil ulang supaya mampu dijalankan pada mesin yang berbeda. Inti dari J2ME terletak pada configuration dan profile-profile. Suatu configuration menggambarkan lingkungan runtime dasar dari suatu sistem J2ME. Ia menggambarkan core library, virtual machine, fitur keamanan dan jaringan.



Gambar 2.2. Arsitektur J2ME

Sebuah profile memberikan library tambahan untuk suatu kelas tertentu pada sebuah perangkat. profile-profile menyediakan user interface(UI) API,persistence, messaging library, dan sebagainya. Satu set library tambahan atau package tambahan menyediakan kemampuan program tambahan. Pemasukan package ini ke dalam perangkat J2ME dapat berubah-ubah karena tergantung pada kemampuan sebuah perangkat. Sebagai contoh, beberapa perangkat MIDP tidak memiliki Bluetooth built-in, sehingga Bluetooth API tidak disediakan dalam perangkat ini.

2.2.1.1. Configuration

Suatu configuration menggambarkan fitur minimal dari lingkungan lengkap Java runtime. Untuk menjamin kemampuan portabilitas dan interoperabilitas optimal diantara berbagai macam perangkat yang dibatasi sumber dayanya(memory, prosesor, koneksi yang dibatasi), configuration tidak menggambarkan fitur tambahan. Suatu configuration J2ME menggambarkan suatu komplemen yang minimum dari teknologi JAVA. Adalah merupakan tugas profile-profile untuk menggambarkan tambahan library untuk suatu kategori perangkat tertentu.

configuration menggambarkan:

- Subset bahasa pemrograman JAVA
- Kemampuan Java Virtual Machine(JVM)
- Core platform libraries
- Fitur sekuriti dan jaringan

2.2.1.2. Profile

Suatu profile menggambarkan set-set tambahan dari API dan fitur untuk pasar tertentu, kategori perangkat atau industri. Sementara configuration menggambarkan library dasar, profile-profile menggambarkan library yang penting untuk membuat aplikasi-aplikasi efektif. Library ini memasukkan user interface, jaringan dan penyimpanan API.

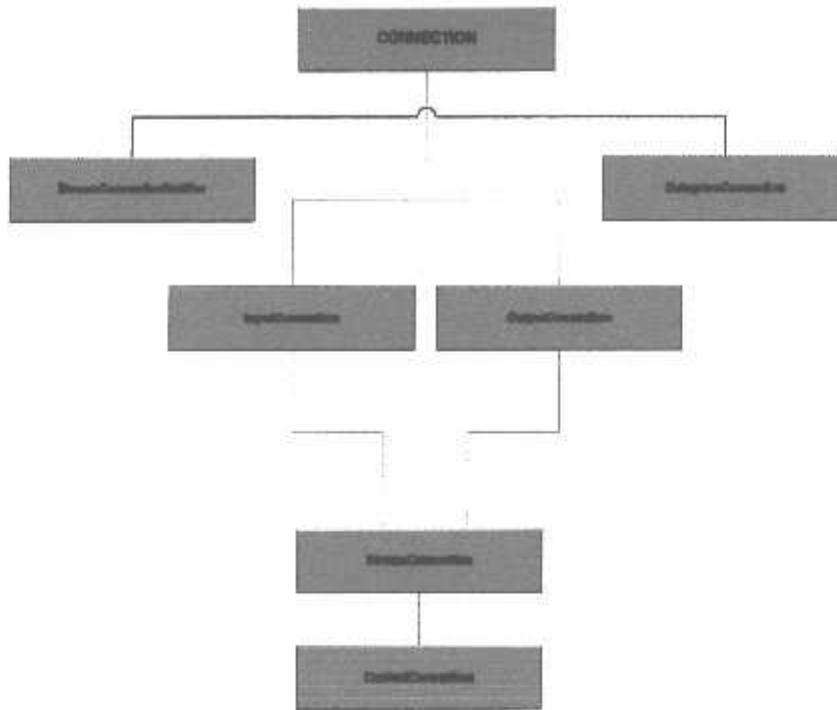
2.2.1.3. CLDC

The Connected Limited Device Configuration (CLDC) menggambarkan dan menunjuk pada area berikut ini:

- Fitur Bahasa Java dan Virtual Machine(VM)
- Library dasar(java.lang.* java.util.*)
- Input/Output(java.io.*)
- Keamanan
- Jaringan
- Internationalization

2.2.1.4. Generic Connection Framework

The Generic Connection Framework menyediakan API dasar untuk koneksi dalam CLDC. Framework ini menyediakan suatu pondasi umum untuk koneksi seperti HTTP, Socket, dan Datagrams. GCF menyediakan suatu set API yang umum dan biasa yang memisahkan semua jenis koneksi. Perlu dicatat bahwa tidak semua jenis koneksi dibutuhkan untuk diterapkan oleh perangkat MIDP. Hirarki interface yang dapat diperluas dari GFC membuat proses penyamarataan menjadi mungkin. Jenis koneksi baru mungkin bisa ditambahkan ke dalam framework ini dengan memperluas hirarki ini.



Gambar 2.3. Hirarki Koneksi GCF (Generic Connection Framework)

2.2.1.5. MIDP

The Mobile Information Device Profile (MIDP) MIDP menggambarkan model aplikasi, UI API, penyimpanan dan jaringan yang kuat, permainan dan media API, kebijakan keamanan, penyebaran aplikasi dan ketentuan over-the-air.

Spesifikasi MIDP menggambarkan suatu perangkat MID yang memiliki karakteristik karakteristik

ini sebagai batas minimum:

- ❖ Tampilan:
 - Ukuran Layar: 96x54
 - kedalaman tampilan: 1-bit

Ketajaman pixel: sekitar 1:1

❖ Masukan:

Satu atau lebih mekanisme user-input: satu keyboard, dua keyboard, atau touch screen

❖ Memory:

256 kilobytes of non-volatile memory untuk implementasi MIDP.

8 kilobytes of non-volatile memory for application-created persistent data

128 kilobytes of volatile memory for the Java runtime (e.g., the Java heap)

❖ Jaringan:

dua jalur, wireless, bandwidth terbatas

❖ Sound:

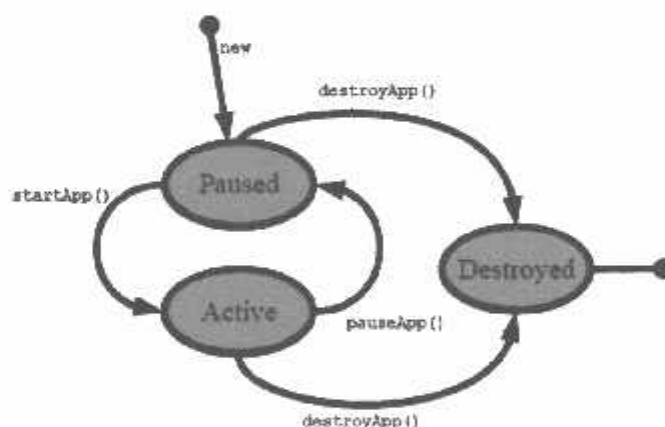
Kemampuan untuk memainkan nada-nada

2.2.1.6.Midlet

Suatu aplikasi MIDP disebut MIDlet. Perangkat application management software (AMS) berinteraksi langsung dengan MIDlet dengan method MIDlet create, start, pause, dan destroy. MIDlet adalah bagian dari package javax.microedition.midlet. Sebuah MIDlet harus di-extend dengan class MIDlet. Dan dapat meminta parameter dari AMS seperti dirumuskan dalam application descriptor (JAD). Suatu MIDlet tidak harus memiliki (dan memang harus tidak mempunyai) sebuah method public static void main(String[] argv).Method tersebut tidak akan dikenal lagi oleh AMS sebagai titik awal sebuah program.

2.2.1.7.Siklus Midlet

Kehidupan MIDlet dimulai ketika di-instantiate oleh AMS. MIDlet pada awalnya masuk status "Pause" setelah perintah baru dibuat. AMS memanggil constructor public tanpa argumen dari MIDlet. Jika sebuah exception terjadi dalam constructor, MIDlet memasuki status "Destroyed" dan membuangnya segera. MIDlet masuk ke dalam status "Active" atas pemanggilan method `startUp()` oleh AMS. MIDlet masuk ke dalam status "Destroyed" ketika AMS memanggil method `destroyApp()`. Status ini juga kembali diakses ketika method `notifyDestroyed()` kembali dengan sukses kepada aplikasi. Dengan catatan bahwa MIDlet hanya bisa memasuki status "Destroyed" sekali dalam masa hidupnya (Antonius Aditya Hartanto *Pemograman Mobile Java dengan MIDP 2.0* Penerbit Andi Yogyakarta).



Gambar 2.4. Siklus Hidup Midlet

2.3. Web Server Apache

Server HTTP Apache atau Server Web/WWW Apache adalah server web yang dapat dijalankan di banyak sistem operasi (Unix, BSD, Linux, Microsoft Windows dan Novell Netware serta platform lainnya) yang berguna untuk

melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan HTTP. Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigur, autentikasi berbasis basis data dan lain-lain. Apache juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI) yang memungkinkan penanganan server menjadi mudah. Apache merupakan perangkat lunak sumber terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan Apache Software Foundation.

Pada awal mulanya, Apache merupakan perangkat lunak sumber terbuka yang menjadi alternatif dari server web Netscape (sekarang dikenal sebagai Sun Java Sistem Web Server). Sejak April 1996 Apache menjadi server web terpopuler di internet. Pada Mei 1999, Apache digunakan di 57% dari semua web server di dunia. Pada November 2005 persentase ini naik menjadi 71%. (sumber: Netcraft Web Server Survey, November 2005). Asal mula nama Apache berasal ketika sebuah server web populer yang dikembangkan pada awal 1995 yang bernama NCSA HTTPd 1.3 memiliki sejumlah perubahan besar terhadap kode sumbernya (patch). Karena banyaknya patch pada perangkat lunak tersebut sehingga disebut sebuah server yang memiliki banyak patch ("a patchy" server). Tetapi pada halaman FAQ situs web resminya, disebutkan bahwa "Apache" dipilih untuk menghormati suku asli Indian Amerika Apache (Indé), yang dikenal karena keahlian dan strategi perangnya. Versi 2 dari Apache ditulis dari awal tanpa mengandung kode sumber dari NCSA (www.wikihost.org).

2.4. PHP

adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memrogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain.

Contoh terkenal dari aplikasi PHP adalah phpBB dan MediaWiki (software di belakang Wikipedia). PHP juga dapat dilihat sebagai pilihan lain dari ASP.NET/C#/VB.NET Microsoft, ColdFusion Macromedia, JSP/Java Sun Microsystems, dan CGI/Perl. Contoh aplikasi lain yang lebih kompleks berupa CMS yang dibangun menggunakan PHP adalah Mambo, Joomla!, Postnuke, Xaraya, dan lain-lain.

PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (Form Interpreted), yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data form dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI, kependekan dari *Hypertext Preprocessing/Form Interpreter*. Dengan perilisannya kode sumber ini menjadi *open source*, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0.

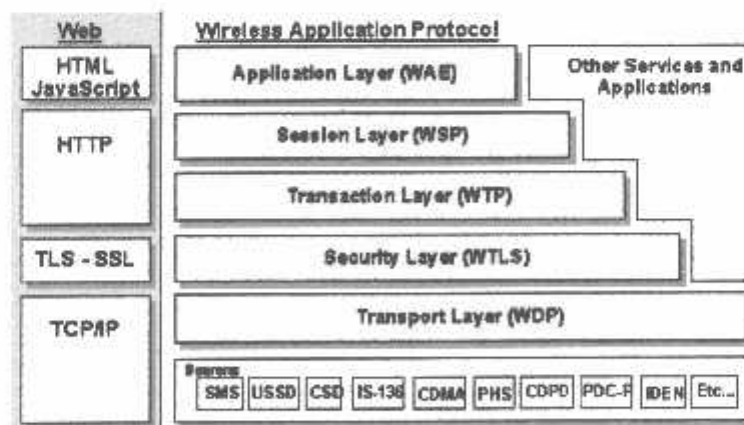
Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai

pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek (www.wikihost.org).

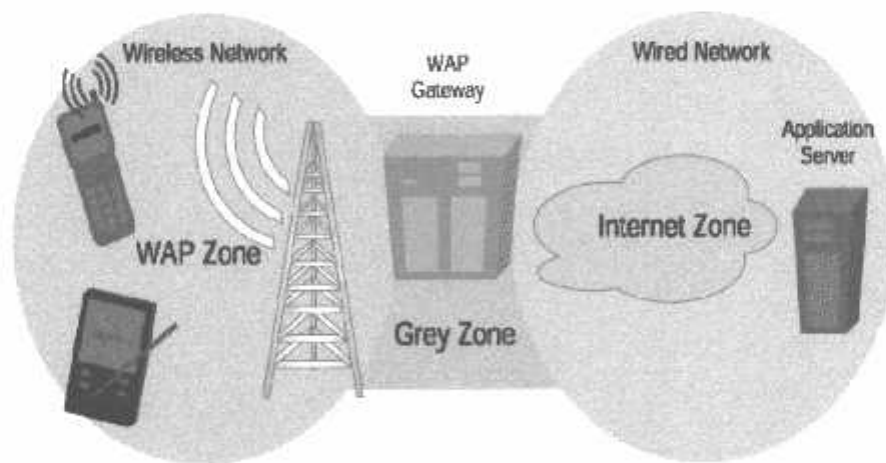
2.5. WAP

Wireless Application Protocol disingkat WAP adalah standar internasional terbuka untuk aplikasi yang menggunakan komunikasi nirkabel. Tujuan utamanya untuk membangun aplikasi yang dapat mengakses internet dari telepon genggam atau PDA. Ada beberapa versi WAP antara lain WAP 1.2.1 dan 2.0. WAP 1.2.1 hanya dapat menampilkan laman sederhana saja dibandingkan dengan WAP 2.0 yang mendukung bahasa XHTML dan gambar. WAP di buat pertama kali sebagai protokol komunikasi bergerak yang tidak bergantung pada sistem tertentu. WAP dirancang sebagai bagian dari sistem di masa depan sama halnya dengan Bluetooth dan GPRS. WAP merupakan protokol komunikasi bergerak yang terdiri dari beberapa layer dan dapat dijalankan pada sistem jaringan yang berbeda.



Gambar 2.5. Layer WAP

WAP sendiri terdiri dari 5 layer, yaitu Wireless Application Environment (WAE), Wireless Session Protocol (WSP), Wireless Transport Protocol (WTP), Wireless Data Protocol (WDP) dan Wireless Transport Layer Security (WTLS). Dari kelima layer tersebut yang terlihat hanyalah WAE, yang menampilkan isi web ke layar. WAE terdiri atas wireless markup language, sebuah variant HTML yang biasa digunakan untuk menampilkan isi web di monitor. WML terdiri dari text dan hyperlink, tetapi bukan grafis. Ketika halaman WAP terlihat di mini browser pada ponsel, bagian lain dari WAP menjalankan bagian peranan penting di belakang layar. Wireless Session Protocol membuka dan menutup koneksi ke situs WAP. Wireless Transport Protocol membuat paket data berjalan sesuai dengan arahnya. Koneksi wireless kurang dapat diandalkan dibandingkan dengan koneksi kabel. Sehingga untuk menjaga agar data yang anda kirim dan terima berjalan dengan benar sangatlah penting. Wireless Layer Security, sebuah bagian dari Secure Socket layer yang biasanya digunakan untuk keamanan di web, mengompres dan mengenkripsi data yang dikirim dari piranti wireless.



Gambar 2.6. Cara Kerja WAP

Saat kita terkoneksi ke jaringan wireless dan meminta akses ke sebuah situs web yang mendukung WAP, telepon seluler akan mengirimkan permintaan tersebut ke gelombang radio ke sel terdekat dan langsung dirutekan ke internet melalui sebuah server gateway. Server gateway tersebut menerjemahkan permintaan ke format standard HTTP dan meneruskannya ke situs web. Jika situs tersebut merespon, maka ia akan mengirimkan dokumen HTML ke server gateway kemudian dikoneksikan ke WML dan merutekannya ke antena terdekat. Antena tersebut akan mengirim via gelombang radio ke piranti WAP anda dan akhirnya micro browser akan menampilkan halaman tersebut (www.IlmKomputer.com).

2.6. GPRS

GPRS merupakan sistem transmisi berbasis paket untuk GSM yang menggunakan prinsip 'tunnelling'. Ia menawarkan laju data yang lebih tinggi. Laju datanya secara kasar sampai 160 kbps dibandingkan dengan 9,6kbps yang dapat disediakan oleh rangkaian tersakelar GSM. Kanal-kanal radio ganda dapat dialokasikan bagi seorang pengguna dan kanal yang sama dapat pula digunakan secara berbagi ('sharing') di antara beberapa pengguna sehingga menjadi sangat efisien. Dari segi biaya, pentarifan diharapkan hanya mengacu pada volume penggunaan. Penggunanya ditarik biaya dalam kaitannya dengan banyaknya byte yang dikirim atau diterima, tanpa memperdulikan panggilan, dengan demikian dimungkinkan GPRS akan menjadi lebih cenderung dipilih oleh pelanggan untuk mengaksesnya daripada layanan-layanan IP. GPRS merupakan teknologi baru yang memungkinkan para operator jaringan komunikasi bergerak menawarkan

layanan data dengan laju bit yang lebih tinggi dengan tarif rendah ,sehingga membuat layanan data menjadi menarik bagi pasar massal. Para operator jaringan komunikasi bergerak di luar negeri kini melihat GPRS sebagai kunci untuk mengembangkan pasar komunikasi bergerak menjadi pesaing baru di lahan yang pernah menjadi milik jaringan kabel, yakni layanan internet. Kondisi ini dimungkinkan karena ledakan penggunaan internet melalui jaringan kabel (telepon) dapat pula dilakukan melalui jaringan bergerak. Sebagai gambaran kecil, layanan bergerak yang kini menjadi sukses di pasar (bagi operator di manca negara) misalnya adalah, laporan cuaca, pemesanan makanan, berita olah raga sampai ke informasi seperti berita-berita penting harian.

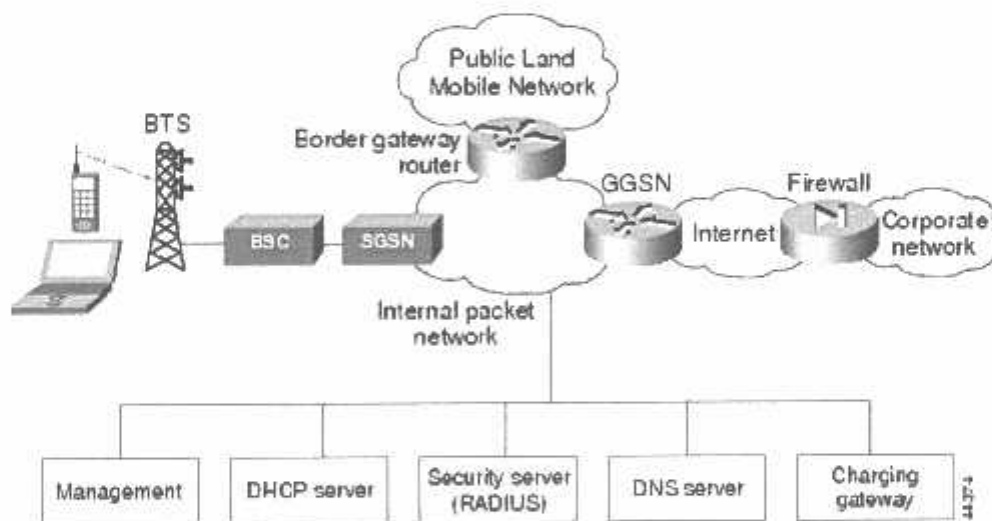
Dalam teorinya GPRS menjanjikan kecepatan mulai dari 56 kbps sampai 115 kbps, sehingga memungkinkan akses internet, pengiriman data multimedia ke komputer, *notebook* dan *handheld computer*. Namun, dalam implementasinya, hal tersebut sangat tergantung faktor-faktor sebagai berikut:

- Konfigurasi dan alokasi time slot pada level BTS
- Software yang dipergunakan
- Dukungan fitur dan aplikasi ponsel yang digunakan

Ini menjelaskan mengapa pada saat-saat tertentu dan di lokasi tertentu akses GPRS terasa lambat, bahkan lebih lambat dari akses CSD yang memiliki kecepatan 9,6 kbps.

Komponen-komponen utama jaringan GPRS adalah :

- GGSN (*Gateway GPRS Support Node*): gerbang penghubung jaringan GPRS ke jaringan internet. Fungsi dari komponen ini adalah sebagai interface ke PDN (Public Data Network), *information routing, network screening, user screening, address mapping.*
- SGSN (*Serving GPRS Support Node*): gerbang penghubung jaringan BSS/BTS ke jaringan GPRS. Komponen ini berfungsi untuk mengantarkan paket data ke MS, *update* pelanggan ke HLR, registrasi pelanggan baru.
- PCU : komponen di level BSS yang menghubungkan terminal ke jaringan GPRS



Gambar 2.7. Cara Kerja GPRS

SGSN bertugas : 1. Mengirim paket ke Mobile Station (MS) dalam satu area 2. Mengirim sejumlah pertanyaan ke HLR untuk memperoleh profile data pelanggan GPRS (*management mobility*) 3. Mendeteksi MS-GPRS yang baru dalam suatu

area servis yang menjadi tanggung jawabnya (location management) 4. SGSN dihubungkan ke BSS pada GSM dengan koneksi *Frame Relay* melalui PCU (Packet Control Unit) di dalam BSC

GGSN bertugas : 1. Sebagai *interface* ke jaringan IP external seperti : public internet atau mobile service provider 2. Meng-*update* informasi *routing* dari PDU (Protokol Data Units) ke SGSN.

GPRS menggunakan sistem komunikasi *packet switch* sebagai cara untuk mentransmisikan datanya. *Packet switch* adalah sebuah sistem di mana data yang akan ditransmisikan dibagi menjadi bagian-bagian kecil (paket) lalu ditransmisikan dan diubah kembali menjadi data semula. Sistem ini dapat mentransmisikan ribuan bahkan jutaan paket per detik. Transmisi dilakukan melalui PLMN (*Public Land Mobile Network*) dengan menggunakan IP *backbone*. Karena memungkinkan untuk pemakaian kanal transmisi secara bersamaan oleh pengguna lain maka biaya akses GPRS, secara teori, lebih murah daripada biaya akses CSD.

GPRS didesain untuk menyediakan layanan transfer packet data pada jaringan GSM dengan kecepatan yang lebih baik dari GSM. Kecepatan yang lebih baik ini didapat dengan menggunakan *coding scheme* (CS) yang berbeda dari GSM (www.IllmuKomputer.com).

BAB III

DESAIN DAN ANALISA SISTEM WEB SINDIKASI

Dalam bab ini akan dibahas desain dan analisa sistem. Analisa akan dilakukan pada setiap blok desain, Desain disini yang dibuat adalah desain dari mobile application dalam melakukan kegiatan sindikasi web. Server meliputi database, dan server penerima request dari client, sedangkan client meliputi content Web Sindikasi, analisa masing-masing blok desain, secara garis besar terdapat dua bagian analisa yaitu : Analisa secara Hardware, Analisa secara Software.

Pada analisa secara hardware meliputi cara kerja komunikasi dari handphone hingga ke server secara hardware pada setiap blok desain, sedangkan analisa secara software meliputi algoritma & tahapan program yang digunakan pada setiap blok desain.

3.1. Spesifikasi Sistem

Sistem Web Sindikasi yang dibangun melakukan komunikasi antara hardware (Handphone) dimana di dalam handphone di install aplikasi Web Sindikasi yang di desain, dari aplikasi itulah user melakukan komunikasi dengan web server, melakukan berbagai kegiatan sesuai dengan konten yang ada, server akan melakukan proses pencarian dan melakukan replay ke user. Maka adapun spesifikasi hardware dan software pada system Web Sindikasi

3.1.1. Spesifikasi Software

Dalam melakukan desain Web Sindikasi di butuhkan beberapa software pendukung yang mampu melengkapi dan mendukung kinerja Web Sindikasi yang di desain hanya pada sisi client saja. Client side (Development application & Testing application).

.

➤ Client Side

Development Application

Dalam mendesain mobile Web Sindikasi di butuhkan software pembangun aplikasi yang terdiri dari :

- J2ME (Java 2 Micro Edition)

Lingkungan pembangun java yang di gunakan adalah J2ME dimana selain terdapat tool untuk develop aplikasi java juga terdapat Java Runtime Environment yaitu sebuah lingkungan untuk menjalankan aplikasi java

- XML Parser (eXtensible Markup Language)

Ada dua cara utama untuk melakukan parsing dokumen XML, yakni dengan *event based parsing* dan *tree based parsing* SAX (*Simple API for XML*) merupakan standart API (*Application Programming Interface*) yang digunakan untuk memarsing XML menggunakan mekanisme *event based parsing*. Mekanisme ini secara sederhana digambarkan sebagai berikut, parser akan membaca file XML secara serial dan setiap saat aplikasi dapat mengetahui tag dan data XML apa yang dibaca oleh parser XML dengan memanggil *callback function*. Keuntungan dari mekanisme ini adalah hemat memory. DOM (*Document Object Model*) menggunakan pendekatan berbeda, yakni

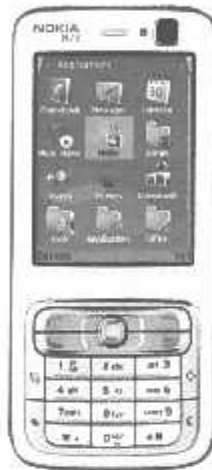
menggunakan mekanisme *Tree based parsing*. Mekanisme ini secara sederhana digambarkan sebagai berikut, parser akan membaca file XML dan membangun struktur pohon dari dokumen XML untuk disimpan ke memory. Aplikasi ini akan membaca struktur data XML yang sudah dalam memori sehingga pemrosesan lebih mudah dan cepat, namun boros memori karena isi dokumen XML untuk disimpan ke memori. Dalam lingkungan J2ME yang terbatas kapasitasnya memorinya, penggunaan DOM jelas tidak efektif, jadi yang lebih efektif adalah menggunakan SAX dalam melakukan parsing document XML.

- NetBeans

NetBeans mengacu pada dua hal, yakni platform untuk pengembangan aplikasi desktop java, dan sebuah Integrated Development Environment (IDE) yang dibangun menggunakan platform NetBeans. Platform NetBeans memungkinkan aplikasi dibangun dari sekumpulan komponen perangkat lunak modular yang disebut 'modul'. Sebuah modul adalah suatu arsip Java (Java archive) yang memuat kelas-kelas Java untuk berinteraksi dengan NetBeans Open API dan file manifestasi yang mengidentifikasinya sebagai modul. Aplikasi yang dibangun dengan modul-modul dapat dikembangkan dengan menambahkan modul - modul baru. Karena modul dapat dikembangkan secara independen, aplikasi berbasis platform NetBeans dapat dengan mudah dikembangkan oleh pihak ketiga secara mudah dan powerful.

Testing Application

Setelah dilakukan pembangunan software maka dibutuhkan emulator untuk testing aplikasi sebelum kita testing pada hardware yang sesungguhnya ini dilakukan agar jika terjadi kesalahan error program tidak sampai mengganggu kinerja hardware, dan kesalahan bisa di tangani lebih awal, emulator yang digunakan sesuai dengan mobile device yang nantinya akan di gunakan yaitu Nokia N73 yang sudah include pada Nokia J2ME(TM) SDK v6.1.1



Gambar 3.1. Nokia N73

3.1.2. Spesifikasi Hardware

Setelah aplikasi yang dibuat telah selesai maka nantinya dilakukan implementasi pada hardware yang sesungguhnya, hardware yang di pakai adalah Handphone Nokia N73 dimana Handphone tersebut kita install aplikasi berekstensi `.jar`, adapun spesifikasi untuk lingkungan JAVA pada Nokia N73 sebagai berikut :

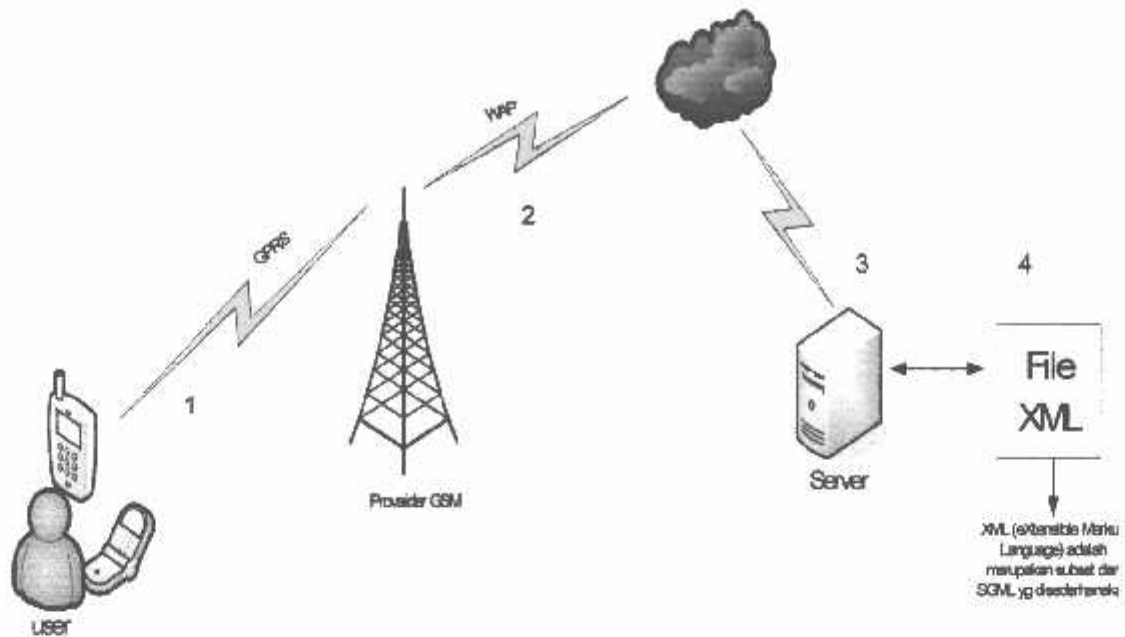
- Java™ MIDP 2.0, CLDC 1.1 (Connected Limited Device Configuration (J2ME))
- C++ dan Java SDKs



Gambar 3.2. Nokia N73

3.2. Desain Keseluruhan Sistem

Perancangan desain Web Sindikasi agar dapat dilakukan secara sistematis dan terstruktur maka perlu dibuat blok sistem yang menjelaskan dari sistem yang dirancang. Secara garis besar perancangan desain sistem ditunjukkan pada gambar berikut :



Gambar 3.3. Desain Keseluruhan Sistem

Keterangan :

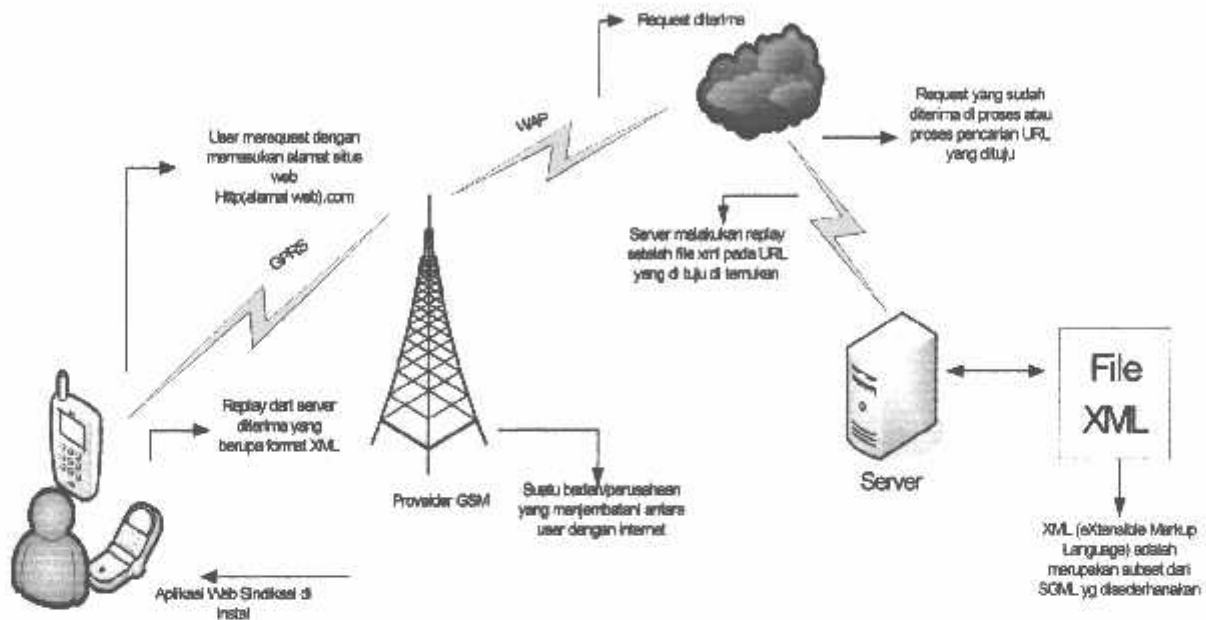
1. Aplikasi User yang di install pada handphone user.
2. Provider GSM yang menterjemahkan sinyal GPRS menjadi request HTTP.
3. Protokol yang digunakan untuk berkomunikasi dengan Internet.
4. Server bersistem oprasi Windows XP, dengan Web Server Apache.
5. XML parsing proses pencacahan agar dapat diterjemahkan ke divice yang lain.

3.2.1. Desain Komunikasi HTTP

Web Server membuka port 80 untuk berkomunikasi dengan dunia luar, maka dalam melakukan komunikasi dengan web server aplikasi user memanfaatkan port tersebut, dimana request dilakukan dengan memetakan URL yang telah ditanam pada script JAVA dan melakukan request dengan metode GET.

➤ Analisa Hardware

Analisa Hardware ini membahas aliran koneksi secara hardware dimana aliran data melewati beberapa proses sebelum diterima oleh server, adapun aliran sebagai berikut:



Gambar 3.4. Analisa Komunikasi HTTP secara Hardware

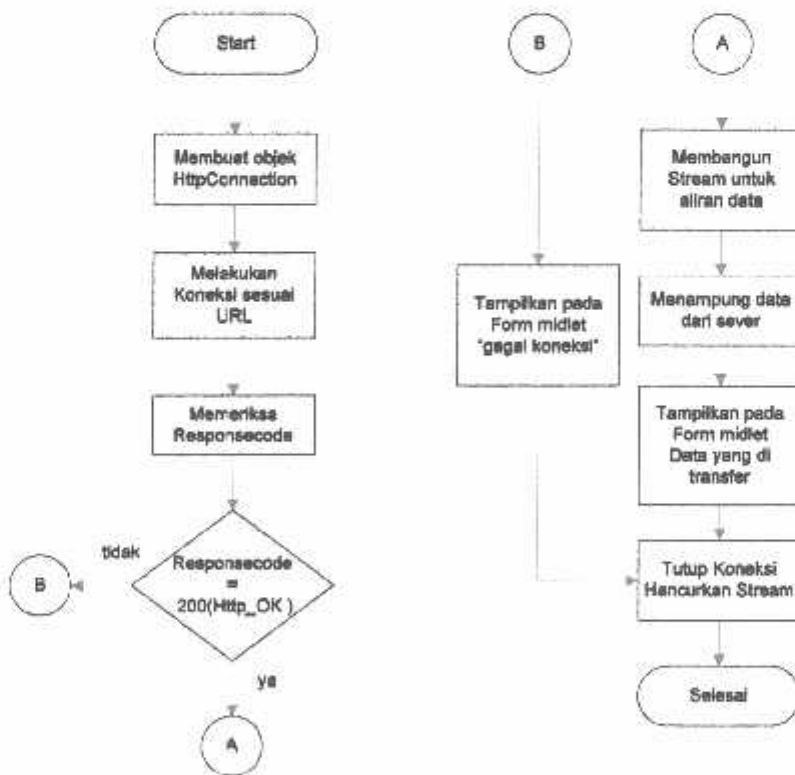
Keterangan dari Desain sistem di atas:

1. User dengan handphone support J2ME di install aplikasi Web Sindikasi, lalu melakukan request ke server, dengan mengakses port HTTP 80 dalam komunikasi dengan server teknologi transfer data yang digunakan pada provider GSM adalah GPRS, awalnya request dari user adalah berupa sinyal GPRS.

2. Pada Provider GSM request dari user yang berupa sinyal GPRS akan di terjemahkan menjadi request berformat HTTP yang nantinya berkomunikasi dengan internet melalui protocol wireless (WAP).
3. Request dari user yang telah di terjemahkan oleh provider ke dalam format HTTP lalu di alamatkan sesuai dengan suatu situs web yang telah ditetapkan dalam aplikasi user, dalam hal ini user melakukan request ke sebuah web server yang websitenya memiliki teknologi RSS (*Really Simple Syndication*) yang berformat XML
4. Server HTTP membuka port 80 untuk berkomunikasi dengan pihak lain, dalam kondisi normal maka server bersikap listening, menunggu paket yang masuk, setelah paket dari user diterima oleh server maka server akan memberikan respon berupa koneksi oke (HTTP_OK 200), setelah aplikasi menerima respon tersebut maka aplikasi user pada java membentuk hubungan khusus dalam hal ini Stream, untuk aliran data khusus antara user dan server.
5. Lalu server mereturn replay ke user melalui route internet untuk dikembalikan pada alamat pengirim dalam hal ini provider GSM, masih lewat hubungan Stream yang di ciptakan java, lalu dari GSM selanjutnya di terjemahkan kembali dalam bentuk sinyal GPRS dan di terima kembali oleh user, setelah user menerima return dari server maka hubungan stream tadi di hancurkan oleh DestroyApp pada packet JAVA.

➤ **Analisa Software**

Komunikasi dengan web server yang dilakukan oleh aplikasi user dalam hal ini lebih mengacu pada pembangunan software J2ME, dalam melakukan komunikasi dengan server algoritma yang digunakan adalah dengan melemparkan paket ke sebuah server sesuai dengan URL aplikasi setelah itu jika server itu ada maka server tersebut akan memberikan nilai kembalian berupa responsecode jika response code = 200 itu berarti koneksi bisa dilakukan, tetapi jika bukan 200 maka koneksi tidak bisa dilakukan, dalam melakukan koneksi tersebut program aplikasi memanfaatkan API dari java yang support dalam melakukan HttpConnection, dimana koneksi akan diciptakan dari melalui objek untuk mengakses API komunikasi, API inilah yang nantinya menangani komunikasi Http terlihat pada flowchart berikut:



Gambar 3.5. Flowchart Komunikasi Http

Tahapan yang dilakukan dalam melakukan komunikasi dengan web server Http sebagai berikut:

0. Inisialisasi Sistem.
1. Membangun objek dari kelas `HttpConnection`, dimana kelas ini berfungsi untuk mengatur koneksi Http.
2. Membangun koneksi ke suatu alamat sesuai dengan URL (Uniform Locator Resource).
3. Dalam melakukan komunikasi dengan server, maka sebelum melakukan komunikasi sesungguhnya ada yang dinamakan pengecekan untuk mengecek apakah alamat yang di tuju sesuai URL benar-benar ada selain itu juga untuk mengecek kondisi jaringan, biasanya server memberikan respon berupa code untuk menggambarkan kondisi yang ada di server adapun beberapa respon code dari server sebagai berikut :

Tabel 3.1. Response code Koneksi HTTP

Code	Response
200	<u>HTTP OK</u>
203	<u>HTTP NOT AUTHORITATIVE</u>
204	<u>HTTP NO CONTENT</u>
305	<u>HTTP USE PROXY</u>
400	<u>HTTP BAD REQUEST</u>
403	<u>HTTP FORBIDDEN</u>
404	<u>HTTP NOT FOUND</u>
405	<u>HTTP BAD METHOD</u>
406	<u>HTTP NOT ACCEPTABLE</u>
408	<u>HTTP CLIENT TIMEOUT</u>
409	<u>HTTP CONFLICT</u>
410	<u>HTTP GONE</u>
413	<u>HTTP ENTITY TOO LARGE</u>
414	<u>HTTP REQ TOO LONG</u>
415	<u>HTTP UNSUPPORTED TYPE</u>

416	<u>HTTP UNSUPPORTED RANGE</u>
417	<u>HTTP EXPECT FAILED</u>
500	<u>HTTP INTERNAL ERROR</u>
501	<u>HTTP NOT IMPLEMENTED</u>
502	<u>HTTP BAD GATEWAY</u>
503	<u>HTTP UNAVAILABLE</u>
504	<u>HTTP GATEWAY TIMEOUT</u>

4. Jika Response code yang di kirim oleh server bukan 200 maka bisa dikatakan koneksi yang terjadi tidak bisa di lanjutkan karena bermasalah, kita tidak perlu mengetahui apa masalahnya yang jelas jika response code bukan 200 maka koneksi putus dan java membangkitkan error message pada form berupa "Gagal Koneksi", sebaliknya jika response code yang di kirim oleh server adalah 200 maka koneksi bisa dilakukan dan aplikasi java user bisa berkomunikasi dengan web server.
5. Setelah server mengirim response code dan code itu bernilai 200 maka java membuat suatu stream dari kelas **InputStreamReader** yang berfungsi untuk melakukan aliran data khusus koneksi ini saja.
6. Setelah Stream terbentuk maka server menerima request dari user dan memprosesnya, setelah server memproses maka server mengirimkan hasil dari proses, yang mana kiriman dari server tersebut ditampung dalam Buffer yang terdapat dalam **InputStreamReader**.
7. Pemeriksaan terhadap kiriman server, apakah kiriman server berisi data atau tidak.
8. Jika hasil dari server kosong atau tidak berisi apa-apa bisa saja akses terhadap file php tujuan user tidak ada dan pada saat itu akan di bangkitkan error message oleh java, sedangkan jika kiriman dari server berisi data maka ditampilkan pada form midlet.

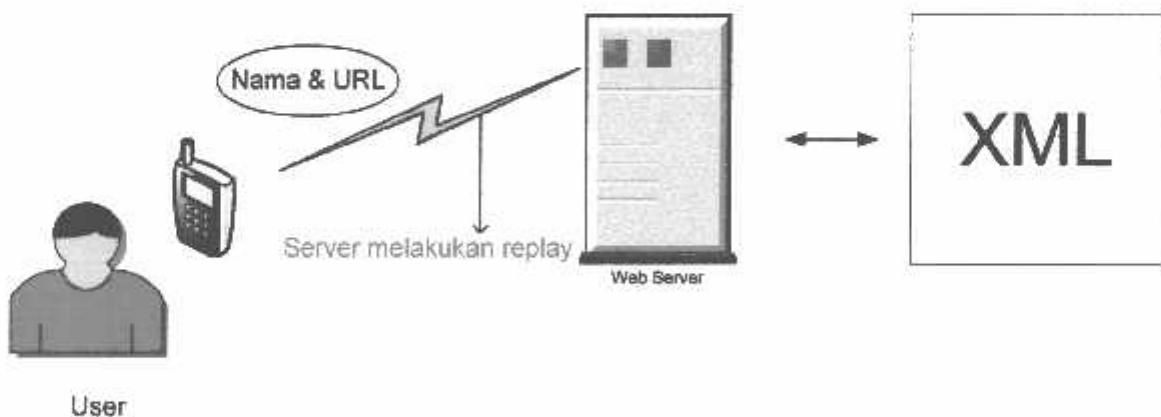
9. Setelah request dari user di terima dan proses pengiriman hasil proses server telah di dapatkan oleh aplikasi user maka hubungan untuk satu koneksi ini di tutup dan stream otomatis di hancurkan.

3.2.2. Desain Sindikasi Web

Dalam melakukan sindikasi web, user melakukan request pada server dan sever akan memproses. seteah suatu situs web yang dituju ditemukan yang langsung merujuk pada umpan web RSS feed yang berformat XML, server akan melakukan replay ke user, adapun system sebagai berikut:

➤ Analisa Hardware

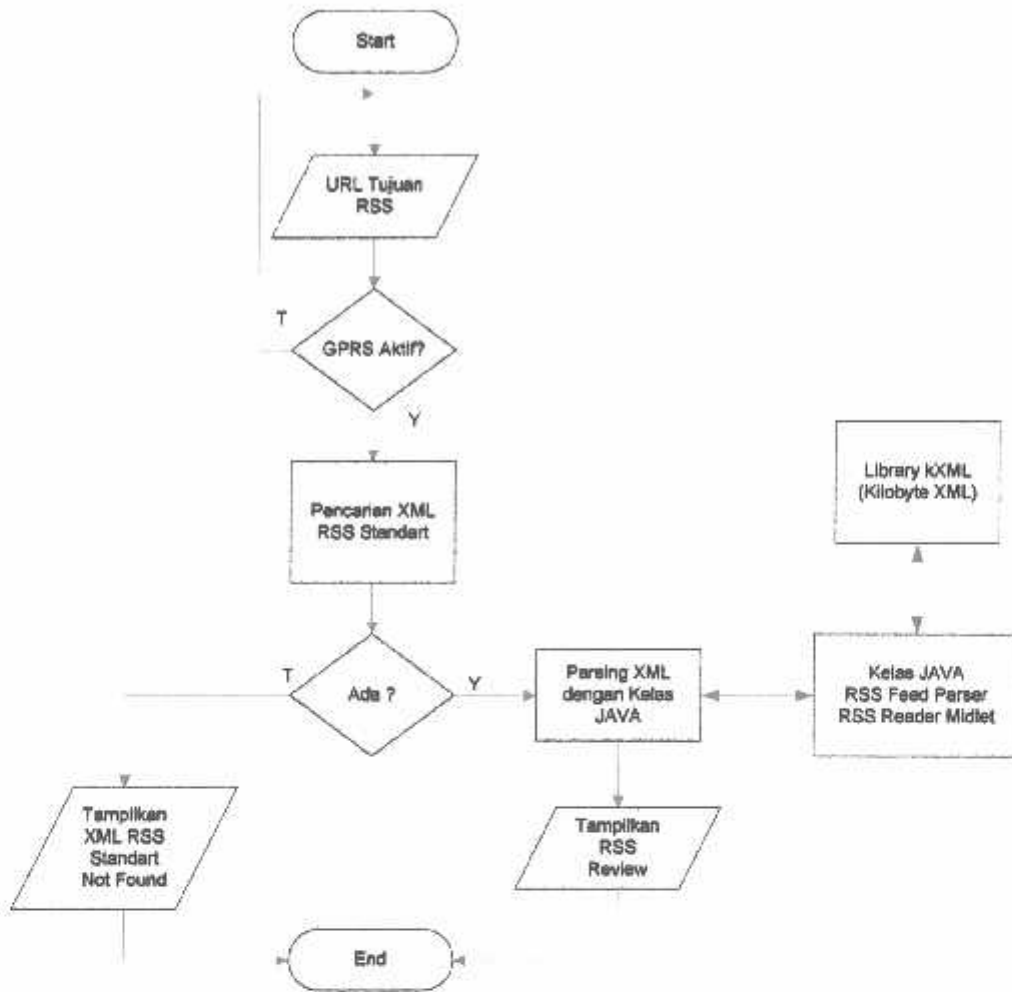
Analisa secara hardware ini menjelaskan bagaimana aliran data secara hardware, adapun desain sebagai berikut:



Gambar 3.6. Request RSS Feed

Keterangan :

1. User melakukan request terhadap server
2. Server menerima request, lalu dilanjutkan ke umpan web RSS yang berformat XML
3. Setelah umpan web RSS berformat XML ditemukan, server melakukan replay terhadap user
4. Aplikasi yang di tanamkan pada mobile melakukan parsing agar dapat ditampilkan, pada mobile ada banyak cara parsing, antara lain menggunakan library J2ME seperti Nano XML, Tiny XML, Kilo XML. Akan tetapi untuk memanfaatkan isi library harus dibuat kelas pada java sehingga dapat tercipta objek untuk memanfaatkan isi library, dan mobile bisa memproses dan menampilkan di layar.



Gambar 3.7. Flowchart Web Sindikasi

Tahapan yang dilakukan dalam melakukan transaksi dengan web server Http sebagai berikut:

0. Inisialisasi sistem.
1. Pengiriman paket nama & URL oleh aplikasi user ke server.
2. Server menerima paket dari user lalu diteruskan ke umpan web RSS yang berformat XML
3. XML melakukan pemrosesan sesuai dengan paket yang masuk
4. Pemeriksaan apakah nama &URL sesuai dengan umpan web yang dituju

5. Jika sesuai maka umpan web yang berformat XML akan ditampilkan , jika tidak sesuai berisi gagal.
6. Aplikasi menerima replay berupa format XML dari server dan aplikasi yang di tanamkan di user akan melakukan parsing agar bias menampilkannya.

Adapun source code inialisai yang ada pada class RssReaderMIDlet.java sebagai berikut

```
RssReaderMIDlet.java
// inialisasi komponen
private Command m_addOkCmd; // The OK command
private Command m_addCancelCmd; // The Cancel command
private Command m_exitCommand; // The exit command
private Command m_addNewBookmark; // The add new bookmark command
private Command m_openBookmark; // The open bookmark command
private Command m_editBookmark; // The edit bookmark command
private Command m_delBookmark; // The delete bookmark command
private Command m_backCommand; // The back to header list command
private Command m_openHeaderCmd; // The open header command
private Command m_backHeaderCmd; // The back to bookmark list command
private Command m_updateCmd; // The update headers command
```

Source code dibawah ini merupakan proses penciptaan dari source code inialisasi diatas

```
RssReaderMIDlet.java
//Penciptaan Objek//
m_addOkCmd = new Command("OK"(*keterangan ini yang tampil di layar),
Command.OK, 1);
m_addCancelCmd = new Command("Cancel", Command.CANCEL, 2);
m_backCommand = new Command("Back", Command.SCREEN, 1);
m_exitCommand = new Command("Exit", Command.SCREEN, 4);
m_addNewBookmark = new Command("Add new", Command.SCREEN, 2);
m_openBookmark = new Command("Open", Command.SCREEN, 1);
m_editBookmark = new Command("Edit", Command.SCREEN, 2);
m_delBookmark = new Command("Delete", Command.SCREEN, 3);
m_openHeaderCmd = new Command("Open", Command.SCREEN, 1);
m_backHeaderCmd = new Command("Back", Command.SCREEN, 2);
m_updateCmd = new Command("Update", Command.SCREEN, 2);
```


Source code dibawah ini merupakan pemanggilan pada tampilan pertama pada aplikasi

```
m_bookmarkList = new List("Bookmarks", List.IMPLICIT);
    m_bookmarkList.addCommand( m_exitCommand );
    m_bookmarkList.addCommand( m_addNewBookmark );
    m_bookmarkList.addCommand( m_openBookmark );
    m_bookmarkList.addCommand( m_editBookmark );
    m_bookmarkList.addCommand( m_delBookmark );
    m_bookmarkList.setCommandListener( this );
```

Source code dibawah ini merupakan source code untuk melakukan conection Http

```
/** send a GET request to web server */
public void parseRssFeed()
throws IOException, Exception {

    HttpURLConnection hc = null;
    DataInputStream dis = null;
    String response = "";
    try {
        /**
         * Open an HttpURLConnection with the Web server
         * The default request method is GET
         */
        hc = (HttpURLConnection) Connector.open( m_rssFeed.getUrl() );
        hc.setRequestMethod(HttpURLConnection.GET);
        /** Some web servers requires these properties */
        hc.setRequestProperty("User-Agent",
            "Profile/MIDP-1.0 Configuration/CLDC-1.0");
        hc.setRequestProperty("Content-Length", "0");
        hc.setRequestProperty("Connection", "close");

        /**
         * Get a DataInputStream from the HttpURLConnection
         * and forward it to kXML parser
         */
    }
```

Source code dibawah ini menjelaskan proses parsing, agar dapat diterjemahkan ke divace yang lain.

```
/**
 * RssFeedParser is an utility class for aquiring and parsing a RSS feed.
 * HttpConnection is used to fetch RSS feed and kXML is used on xml parsing.
 */
public class RssFeedParser {

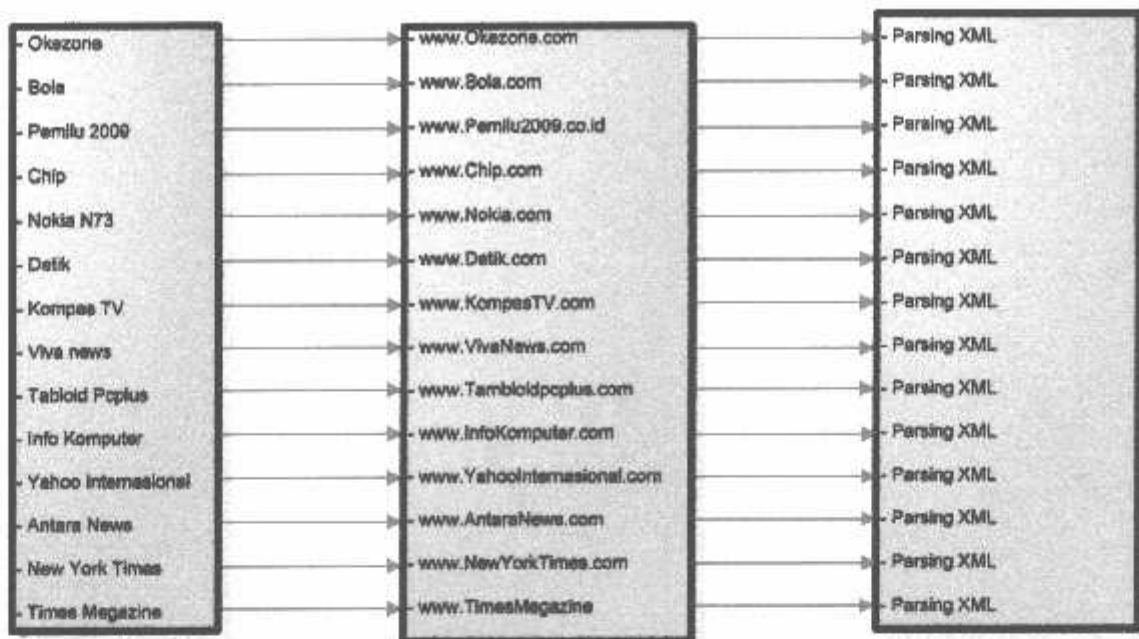
    private RssFeed m_rssFeed; // The RSS feed
    private XmlPullParser m_xmlParser = new KXmlParser(); // The Xml parser

    /** Create new instance of RssDocument */
    public RssFeedParser(RssFeed rssFeed) {
        m_rssFeed = rssFeed;
    }

    /** Return RSS feed */
    public RssFeed getRssFeed() {
        return m_rssFeed;
    }
}
```

3.2.3. Desain Konten

Dalam sebuah aplikasi Web Sindikasi ada hal yang terpenting yang harus di perhatikan yaitu konten, jika kita berbicara konten tentunya konten ini harus sesuai dengan kebutuhan dan keinginan pasar jangan sampai ada konten yang sia-sia saja, oleh karena itu di rancanglah konten aplikasi Web Sindikasi yang dapat memenuhi segala kegiatan untuk mendapatkan informasi terUpdate, dimana dengan konten-konten inilah kegiatan untuk mendapatkan berita terUpdate lebih terorganisir, karena setiap kebutuhan telah di desain sedemikian rupa sehingga memudahkan user dalam melakukan kegiatan untuk mendapatkan informasi terUpdate secara mobile, adapun rancangan konten dalam aplikasi Web Sindikasi sebagai berikut :



Gambar 3.8. Desain Konten Web Sindikasi

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM WEB SINDIKASI

Sebagai cara untuk mengetahui untuk kerja dari sistem yang dirancang dan dibuat, maka dilakukan implementasi dan pengujian sistem, Implementasi meliputi install aplikasi Web Sindikasi pada handphone Nokia N73 Pengujian meliputi koneksi antara aplikasi di handphone saja.

4.1. Pengujian Konten

Pengujian Konten dilakukan dengan menggunakan Handphone Nokia N73 yang telah di install Aplikasi Web Sindikasi, yang mana nantinya dalam konten tersebut telah dipetakan website berita yang akan disindikasi. Konten ini memudahkan user untuk dapat memilih dan membaca berita, tiap konten memiliki fasilitas untuk update sehingga website yang memiliki update sangat cepat bisa terus kita pantau dengan memanfaatkan fasilitas ini.

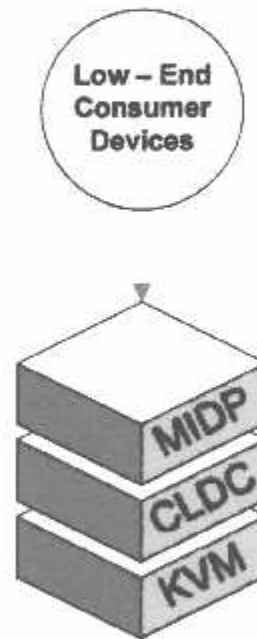


Gambar 4.1. Hasil Content

4.2 Pengujian Komunikasi

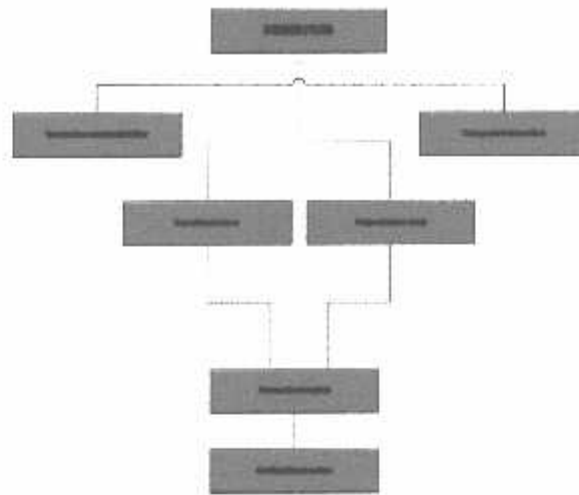
Dalam melakukan koneksi user tinggal memilih alamat website yang telah tersedia pada aplikasi web sindikasi, Maka aplikasi akan membuka koneksi HTTP, dan mentransfer data xml, melakukan request ke web server, dengan metode HEAD untuk mendapatkan informasi header dari sebuah web server, jika kita mendapatkan response maka web server akan mengirimkan tentang informasi terUpdate dari web server, dengan begitu bisa dikatakan bahwa komunikasi antara aplikasi Web Sindikasi dengan web server bisa dilakukan.

Dalam Melakukan request kepada web server, java dalam hal ini J2ME (Low end – Consumer User) menggunakan MIDP versi 2.0 yang mengacu pada display layar yang telah di dukung oleh Nokia N73, CLDC untuk komunikasi jaringan mengacu pada CLDC, dimana dalam CLDC menggambarkan salah satunya jaringan.



Gambar 4.2. Arsitektur Low – End Consumer Devices

API dasar untuk koneksi CLDC disediakan oleh sebuah Framework dimana didalamnya telah tersedia berbagai koneksi datagram, Stream dll framework itu adalah GFC (Generic Connection Framework)

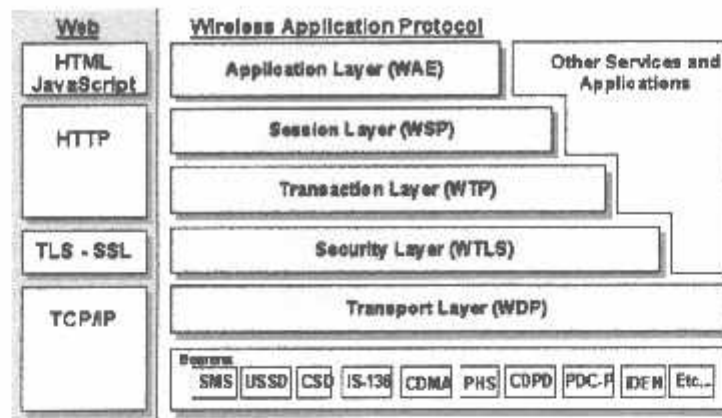


Gambar 4.3. Hirarki GCF

Dalam membuka koneksi kita bisa menggunakan `Connector.open()` dan mengcastingnya dengan salah satu dari ketiga interface berikut : `StreamConnection`, `ContentConnection`, `HttpConnection`, jika pada `StreamConnection` kita bisa menentukan panjang dari sebuah replay, berbeda dengan `HttpConnection` dan `ContentConnection` dimana panjang replay bisa kita tentukan, interface yang digunakan dalam koneksi HTTP berikut adalah `HttpConnection`.

`HttpConnection` memiliki banyak sekali fungsi yang bisa kita manfaatkan, setelah kita rujuk interface `HttpConnection`, maka paket kita lempar ke web server, dalam hirarki perjalannya paket dilempar berupa sinyal GPRS lalu di terjemahkan oleh provider menjadi standart request HTTP dan di teruskan ke web server menggunakan WAP.

Transfer paket yang dilakukan oleh provider ke jaringan global internet melalui WAP (Wireless Application Protokol), dimana paket yang di alirkan melalui Session Protokol yang di gunakan untuk membuka koneksi jaringan, Aliran paket dari jaringan di terima dan dikirim melalui datagram protocol.



Gambar 4.4. Layer WAP

Adapun source code sebagai berikut :

```

public void parseRssFeed()
throws IOException, Exception {

    HttpURLConnection hc = null;
    DataInputStream dis = null;
    String response = "";
    try {
        /**
         * Membuka koneksi Http dengan web server
         * Metode permintaan exception
         */
        hc = (HttpURLConnection) Connector.open(
m_rssFeed.getUrl() );
        hc.setRequestMethod(HttpURLConnection.GET);
        /** Beberapa server web memerlukan property ini */
        hc.setRequestProperty("User-Agent",
            "Profile/MIDP-1.0 Configuration/CLDC-1.0");
        hc.setRequestProperty("Content-Length", "0");
        hc.setRequestProperty("Connection", "close");

        /**
         * Dapatkan suatu datainputstream dari
Httpconnection
         * dan maju untuk ke XML parser
         */
        parseRssFeedXml( hc.openInputStream() );
    } catch(Exception e) {
        throw new Exception("Error while parsing RSS data: "
            + e.toString());
    } finally {
        if (hc != null) hc.close();
        if (dis != null) dis.close();
    }
}
}

```

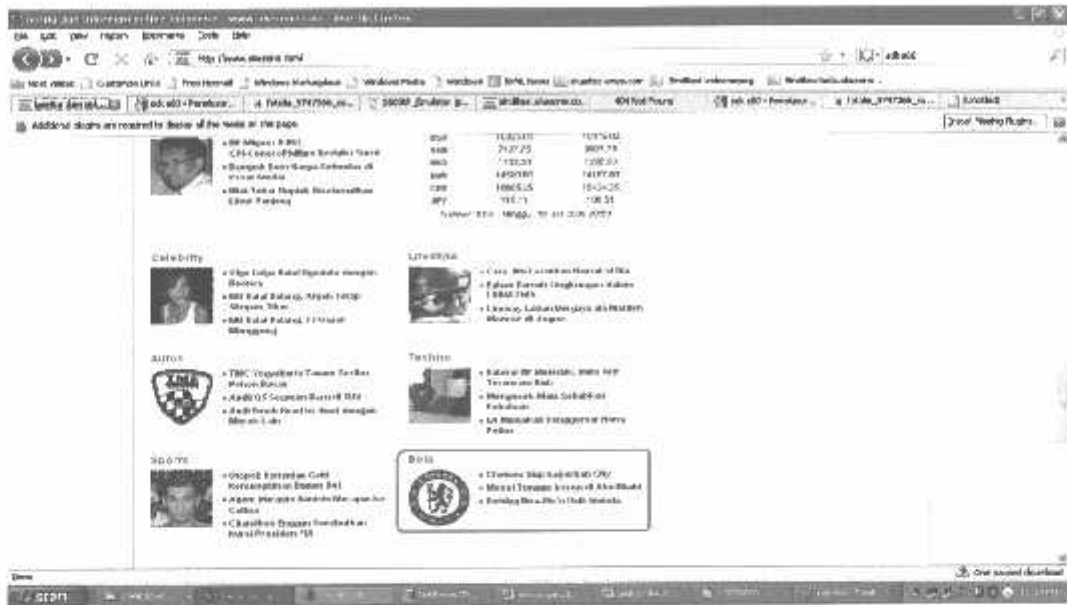
Hasil dari source code diatas adalah sebuah informasi header web server yang memberikan informasi tentang web server yang kita rujuk, dalam hal ini adalah web sindikasi yang berformat XML



Gambar 4.5 Hasil Komunikasi Http

4.2.1 Analisa Pada Website www.bola.com

Untuk bola, pada website resmi okezone hanya ditampilkan 3 berita terupdate. Tapi pada aplikasi dapat tampil selain 3 terupdate, juga berita yang sebelumnya. Pada hari itu



Gambar 4.6 Tampilan Website Pada Komputer Dekstop



Gambar 4.7 Tampilan Pada Aplikasi



Gambar 4.8 Tampilan Website Pada Komputer Dekstop

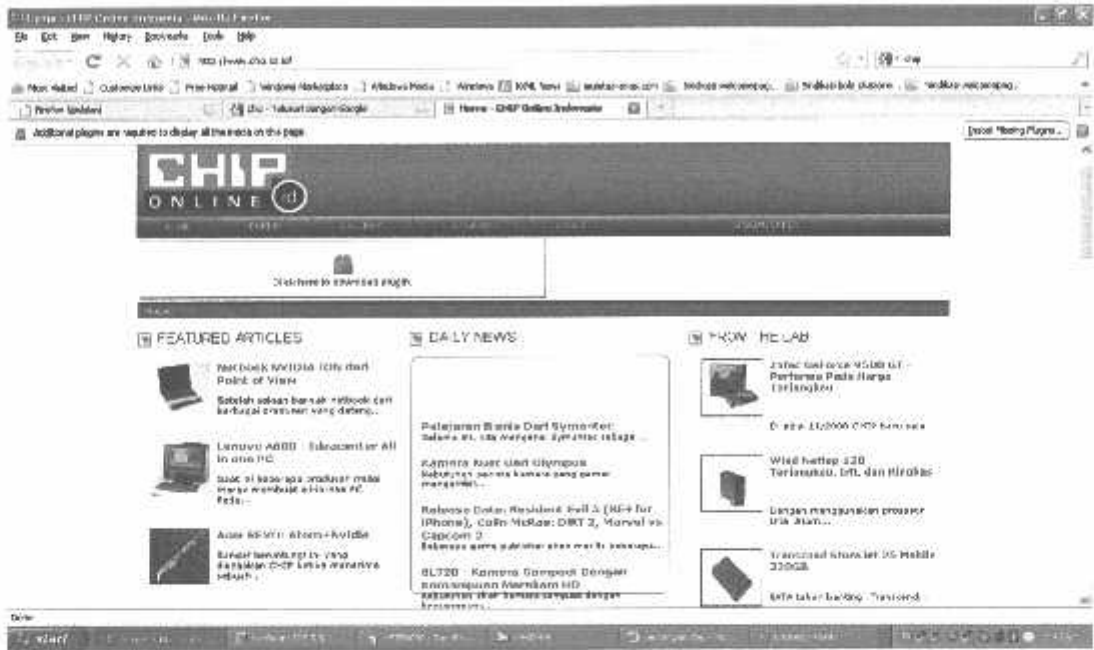
Tampilan pada aplikasi setelah salah satu headline dipilih dan dibuka



Gambar 4.9 Tampilan Pada Aplikasi

4.2.2 Analisa Pada WebSite www.chip.co.id

Pada website telah disediakan suatu hider khusus untuk berita terupdate yang terdapat pada halaman depan



Gambar 4.10 Tampilan Website Pada Komputer Dekstop



Gambar 4.11 Tampilan Pada Aplikasi

Tampilan pada website setelah memilih salah satu headline dari berita terupdate



Gambar 4.12 Tampilan Website Pada Komputer Dekstop

Tampilan pada aplikasi setelah salah satu headline dipilih dan dibuka



Gambar 4.13 Tampilan Pada Aplikasi

4.2.3 Analisa Pada Website www.vivanews.com

Untuk viva news, pada website resmi vivanews hanya ditampilkan 3 berita terupdate. Tapi pada aplikasi dapat tampil selain 3 terupdate, juga berita yang sebelumnya pada hari itu



Gambar 4.14 Tampilan Website Pada Komputer Dekstop



Gambar 4.15 Tampilan Pada Aplikasi

Tampilan pada website setelah memilih salah satu headline dari berita terupdate



Gambar 4.16 Tampilan Website Pada Komputer Dekstop

Setelah salah satu headline pada aplikasi dipilih maka tampilannya akan seperti ini



Gambar 4.17 Tampilan Pada Aplikasi

4.2.4 Tampilan Pada Website www.antarancws.com

Pada website telah disediakan suatu header khusus untuk berita terupdate yang terdapat pada halaman depan, tetapi tidak semua berita terupdate dapat ditampilkan



Gambar 4.18 Tampilan Website Pada Komputer Desktop



Gambar 4.19 Tampilan Pada Aplikasi

Tampilan pada website setelah membuka atau memilih salah satu headline dari berita terupdate



Gambar 4.20 Tampilan Website Pada Komputer Dekstop

Setelah salah satu headline pada aplikasi dipilih maka tampilannya akan seperti ini



Gambar 4.21 Tampilan Pada Aplikasi

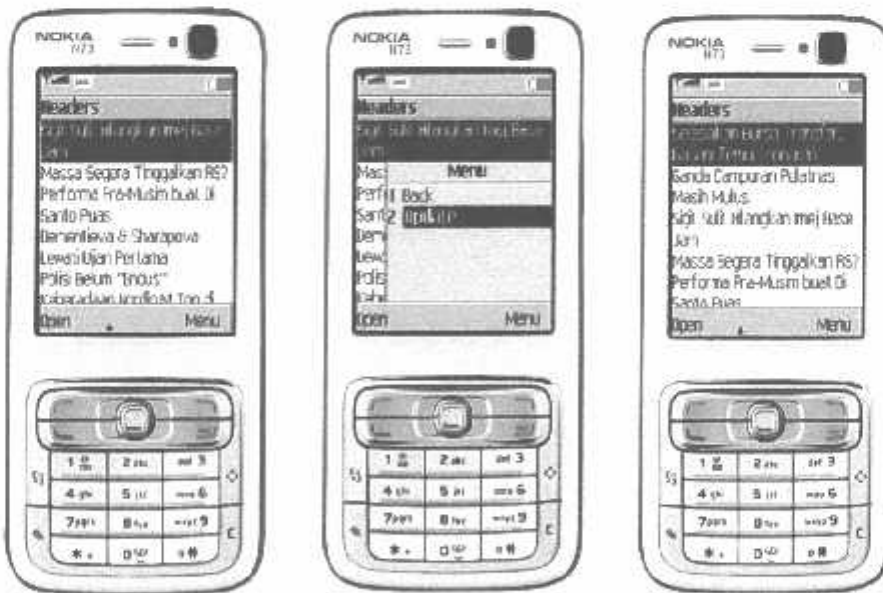
4.3 Analisa Pengujian Update

Pengujian update dilakukan pada saat koneksi dengan web server masih terbuka, jika user ingin melakukan update pada suatu content yang sudah diakses. Maka user cukup memilih menu update pada aplikasi, itu berarti aplikasi melakukan request lagi terhadap web site yang dirujuk.

Source code :

```
private void updateHeaders() {
    initializeLoadingForm();
    m_display.setCurrent( m_loadForm );

    if(m_curRssParser.getRssFeed().getUrl().length()>0) {
        m_getPage = true;
    }
}
```



Gambar 4.22 Sebelum, Sesaat & Sesudah Update

4.4 Analisa Pengujian Kecepatan Koneksi

Pengujian kecepatan koneksi dilakukan untuk mengetahui performa dari aplikasi ini, pengujian dilakukan pada handphone Nokia N73 S60. Pengujian ini di coba dari berbagai macam provider, diantaranya adalah IM3, Telkomsel, Three, XL, Axis. Setelah dilakukan pengujian dalam beberapa hari ini ternyata kecepatan akses bergantung pada jam berapa user menggunakan aplikasi ini, jika pada saat tryfick tinggi user melakukan request, maka koneksi ke web server agak lama. Sebaliknya jika user melakukan request di saat traffic rendah, maka koneksi ke web server akan lebih cepat. Kecepatan koneksi juga bukan hanya pada traffic tinggi atau rendah dan kartu apa yang di gunakan. Akan tetapi kecepatan koneksi juga di pengaruhi oleh handphone yang digunakan.

Table dibawah merupakan hasil percobaan yang dilakukan pada saat tryfick rendah , dan pada saat signal penuh.

Tabel 4.1 Pengujian Kecepatan Koneksi

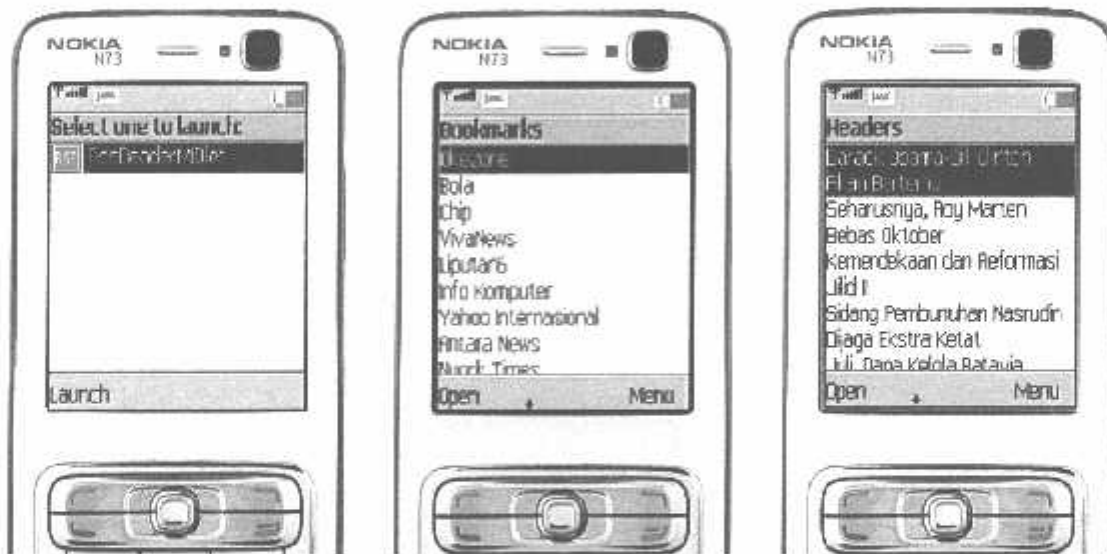
Vendor	Hasil
IM3	8 detik
TELKOMSEL	10 detik
XL	16 detik
3	8 detik
AXIS	6 detik

4.5 Pengujian Aplikasi Web Sindikasi Pada Beberapa Jenis HandPhone

Didalam pengujian ini aplikasi ditanamkan atau diinstal pada beberapa jenis handphone diantaranya adalah Nokia, LG, Samsung, Siemens, dan Sony Ericsson. Syarat utama factor pendukung agar aplikasi web sindikasi ini dapat dijalankan yaitu adanya pendukung GPRS pada handphone tersebut. Dikarenakan aplikasi web sindikasi dibangun dari java yang berekstensi **.jar**, maka aplikasi ini bersifat multiplatform. adapun spesifikasi untuk lingkungan JAVA

- CLDC Version : 1.1
- MIDP Version : 2.0

4.5.1 Tampilan Pengujian pada handphone Nokia



Gambar 4.23 Pengujian Aplikasi

4.5.2 Tampilan Pengujian Pada Handphone LG



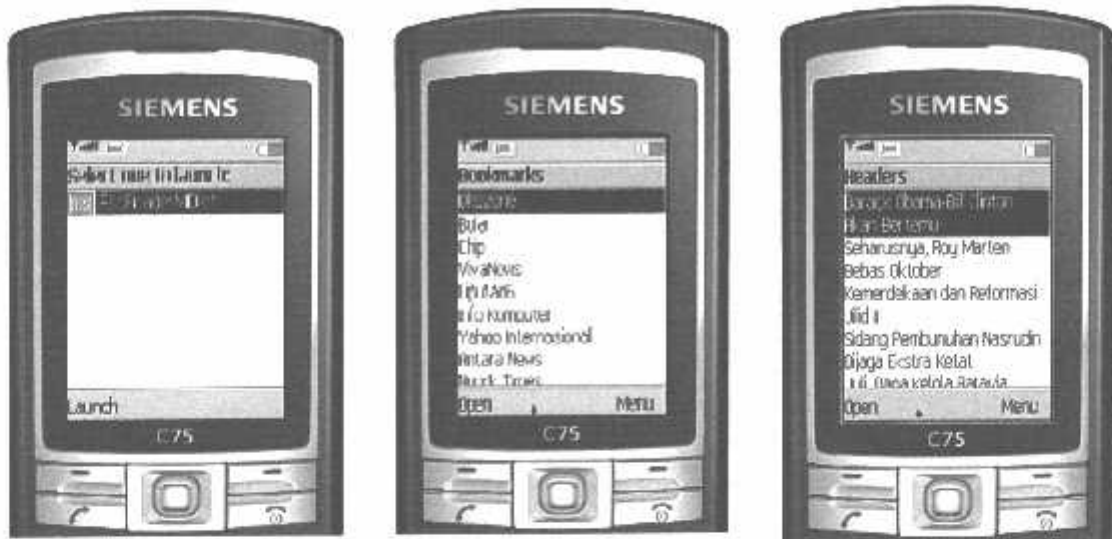
Gambar 4.24 Pengujian Aplikasi

4.5.3 Tampilan Pengujian Pada Handphone Samsung



Gambar 4.25 Pengujian Aplikasi

4.5.4 Tampilan Pengujian Pada Handphone Siemens



Gambar 4.26 Pengujian Aplikasi

4.5.4 Tampilan Pengujian Pada Handphone Sony Ericsson



Gambar 4.27 Pengujian Aplikasi

BAB V

PENUTUP

5.1. Kesimpulan

1. Aplikasi web sindikasi ini mampu berjalan di semua merek handphone, dengan syarat utama handphone tersebut mendukung java dan terkoneksi dengan GPRS, adapun spesifikasi pada lingkungan java yaitu menggunakan CLDC 1.1 dan MIDP 2.0
2. kecepatan akses dalam menjalankan aplikasi ini bergantung pada setiap vendor yang digunakan, sebab masing vendor memberikan layanan yang berbeda – beda pada user
3. Aplikasi web sindikasi lebih cepat mendapatkan berita atau informasi terupdate dari website tertentu, karena aplikasi web sindikasi langsung merujuk pada umpan Rss pada website yang menggunakan teknologi ini.
4. Aplikasi web sindikasi lebih mudah penggunaanya dan mendukung mobilitas, karena untuk mendapatkan berita atau informasi terupdate dari statu web site kita hanya menggunakan handphone, user tidak perlu susah – susah dengan menggunakan labtop ataupun komputer dekstop untuk mendapatkan berita atau informasi terupdate

5.2. Saran

Kelemahan pada sistem aplikasi web sindikasi ini yaitu :

1. Keterbatasan memory handphone menyebabkan lamanya proses jika data yang dikirimkan oleh server untuk mendapatkan berita atau informasi terupdate terlalu besar.
2. User tidak bisa mencari berita yang sudah lama atau sudah kadaluarsa, sebab aplikasi web sindikasi ini cuman melakukan update terhadap server.
3. Aplikasi web sindikasi ini tidak bisa menyimpan berita yang sudah user baca, ini dikarenakan web sindikasi ini tidak dilengkapi dengan data base. Ini bisa dijadikan pengembangan sekripsi orang lain untuk selanjutnya.
4. Aplikasi web sindikasi ini tidak dapat memparsing file gambar jika di server rss ada tampilan gambarnya, aplikasi ini hanya bisa membaca alamat dan nama gambar.

LAMPIRAN





INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Karanglo km 2, Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Iwan Hadi Mulyono
NIM : 04.12.687
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer dan Informatika
Judul Skripsi : **DESAIN SISTEM SINDIKASI WEB MELALUI APLIKASI MOBILE
MENGUNAKAN JAVA (J2ME)**

Dipertahankan di hadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Sabtu
Tanggal : 07 September 2009
Dengan Nilai : 83,45 (A) *84*



Ir. H. Sidik Noertjahjono, MT.
NIP.Y 102 8700 163

Sekretaris Majelis Penguji

Ir. F. Yudi Limpraptono, MT.
NIP.Y 103 9500 274

Penguji I

Joseph Dedy Irawan, ST, MT
NIP.Y 132315178

Penguji II

Setyohadi, ST
NIP.Y. 103 9700 309

INSTITUT TEKNOLOGI NASIONAL
Jl. Sigura-guru No 2
MALANG

Lampiran : 1 (Satu) Berkas
Pembimbing Skripsi

Kepada : Yth. I Komang Sumawirata, ST, MT
Dosen Institut Teknologi Nasional
MALANG

Yang bertanda tangan di bawah ini:

Nama : Iwan Hadi Mulyono
Nim : 04.12.687
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer & Informatika

Dengan ini mengajukan permohonan, kiranya Bapak bersedia menjadi Dosen Pembimbing (Utama / Pendamping *), untuk penyusunan Skripsi dengan judul (proposal terlampir) :


**“DESAIN SISTEM SINDIKASI WEB MELALUI APLIKASI MOBILE
MENGUNAKAN JAVA (J2ME)”**

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Skripsi Sarjana Teknik.

Demikian permohonan kami dan atas kesediaan Bapak/Ibu kami ucapkan terima kasih

Malang, 15 Januari 2009

Ketua
Jurusan Teknik Elektro S-1


Ir. F. Yudi Lampraptono, MT
NIP. Y. 1039500274

*) coret yang tidak perlu

Hormat kami,


Iwan Hadi Mulyono

Form S-3a

INSTITUT TEKNOLOGI NASIONAL
Jl. Sigura-gura No 2
MALANG

Lampiran : 1 (Satu) Berkas
Pembimbing Skripsi

Kepada : Yth. M. Ashar, ST. MT.
Dosen Institut Teknologi Nasional
MALANG

Yang bertanda tangan di bawah ini:

Nama : Iwan Hadi Mulyono
Nim : 04.12.687
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer & Informatika

Dengan ini mengajukan permohonan, kiranya Bapak bersedia menjadi Dosen Pembimbing (~~Utama~~ / Pendamping *), untuk penyusunan Skripsi dengan judul (proposal terlampir) :

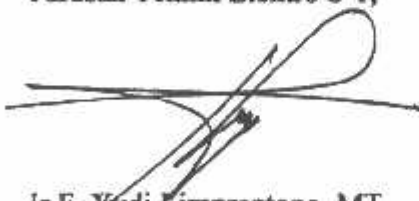
“DESAIN SISTEM SINDIKASI WEB MELALUI APLIKASI MOBILE
MENGUNAKAN JAVA (J2ME)”

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Skripsi Sarjana Teknik.

Demikian permohonan kami dan atas kesediaan Bapak/Ibu kami ucapkan terima kasih.

Malang, 15 Januari 2009

Ketua
Jurusan Teknik Elektro S-1,



Ir. F. Yudi Limpraptono, MT
NIP.Y. 1039500274

Hormat kami,



Iwan Hadi Mulyono

*) coret yang tidak perlu

Form S-3a



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

T. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 851431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 18 Februari 2009

Nomor : ITN-738/I.TA/2/09
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr./i. M. ASHAR, ST, MT
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama : IWAN HADI M
Nim : 0412687
Fakultas : Teknologi Industri
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer & Informatika

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya
kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai
tanggal :

03 Februari 2009 s/d 03 Agustus 2009

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan
terima kasih



Ketua Jurusan
Teknik Elektro S-1

I. F. Yudi Limprantono, MT
Nip. Y. 1039500274

Tembusan Kepada Yth.:

1. Mahasiswa Yang bersangkutan
2. Arsip

Form S 4e

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa :

Nama : Iwan Hadi Mulyono

Nim : 04.12.687

Jurusan : Teknik Elektro S-1

Konsentrasi : Teknik Komputer & Informatika

Dengan ini Menyatakan (bersedia / tidak bersedia *) Membimbing Skripsi dari mahasiswa tersebut, dengan judul :

**“DESAIN SISTEM SINDIKASI WEB MELALUI APLIKASI MOBILE
MENGUNAKAN JAVA (J2ME)”**

Demikian surat Pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, 15 Januari 2009

Kami yang membuat pernyataan,



I Komang Somawirata, ST, MT

Nip.Y.1030100361

Catatan :

Setelah disetujui agar formulir ini
Diserahkan mahasiswa/i yang bersangkutan
Kepada Jurusan untuk diproses lebih lanjut.

***) coret yang tidak perlu**

Form S-3b

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa :

Nama : Iwan Hadi Mulyono

Nim : 04.12.687

Jurusan : Teknik Elektro S-1

Konsentrasi : Teknik Komputer & Informatika

Dengan ini Menyatakan (bersedia / tidak bersedia *) Membimbing Skripsi dari mahasiswa tersebut, dengan judul :

“DESAIN SISTEM SINDIKASI WEB MELALUI APLIKASI MOBILE
MENGUNAKAN JAVA (J2ME)”

Demikian surat Pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, 15 Januari 2009

Kami yang membuat pernyataan,



M. Ashar, ST., MT.
NIP.Y. 1039700309

Catatan :

Setelah disetujui agar formulir ini
Diserahkan mahasiswa/i yang bersangkutan
Kepada Jurusan untuk diproses lebih lanjut.
*) coret yang tidak perlu

Form S-3b



FORMULIR BIMBINGAN SKRIPSI

Nama : IWAN HADI MULYONO
Nim : 04.12.687
Masa Bimbingan : 03 FEBRUARI 2009 s/d 03 AGUSTUS 2009
Judul Skripsi : DESAIN SISTEM SINDIKASI WEB MELALUI APLIKASI MOBILE
MENGUNAKAN JAVA (J2ME)

No.	Tanggal	Uraian	Paraf Pembimbing
1.		Bab III Penulisan Uraian pada menggunakan Analisis Software	
2.		Bab IV Pembahasan Tampilan pada Laporan	
3.			
4.			
5.			
6.			
7.			
8.			
9.			
10.			

Malang,
Desen Pembimbing

I Komang Somawirata, ST, MT.
NIP: Y.1030100361

Form S-4B



FORMULIR BIMBINGAN SKRIPSI

Nama : IWAN HADI MULYONO
Nim : 04.12.687
Masa Bimbingan : 03 FEBRUARI 2009 s/d 03 AGUSTUS 2009
Judul Skripsi : DESAIN SISTEM SINDIKASI WEB MELALUI APLIKASI MOBILE
MENGUNAKAN JAVA (J2ME)

No.	Tanggal	Uraian	Paraf Pembimbing
1.		Bab I, II, III kerangka penulisan.	A
2.		Bab IV kerangka penulisan dimana konsep dilampirkan. sambungkan Analisa pengujian kegunaan Akers jurnal konsep kerangka.	A
3.			A
4.			A
5.			A
6.			A
7.			A
8.			A
9.			A
10.			A

Malang,
Dosen Pembimbing

M. Ashar, ST, MT
NIP : Y.103050048

Form S-4B

LAMPIRAN

```
/**
 * RssFeed.java
 */

package RssReader;

import java.util.*;

/**
 * RssFeed class contains one RSS feed's properties.
 * Properties include name and URL to RSS feed.
 *
 * @author Tommi Laukkänen
 */
public class RssFeed{

    private String m_url = "";
    private String m_name = "";
    protected Vector m_items = new Vector(); // The RSS item vector

    /** Creates a new instance of RSSBookmark */
    public RssFeed(String name, String url){
        m_name = name;
        m_url = url;
    }

    /** Creates a new instance of RSSBookmark with record store string */
    public RssFeed(String storeString){
        int i = storeString.indexOf("|");
        if(i>0) {
            m_name = storeString.substring(0,i);
            m_url = storeString.substring(i+1);
        }
    }

    /** Return bookmark's name */
    public String getName(){
        return m_name;
    }

    /** Return bookmark's URL */
    public String getUrl(){
        return m_url;
    }
}
```

```

    /** Return record store string */
    public String getStoreString(){
        return m_name + "|" + m_url;
    }

    /** Return RSS feed items */
    public Vector getItems() {
        return m_items;
    }
}

/**
 * RssFeedParser.java
 */

package RssReader;

import javax.microedition.io.*;
import java.util.*;
import java.io.*;
import org.xmlpull.v1.*;
import org.kxml2.io.*;

/**
 * RssFeedParser is an utility class for acquiring and parsing a RSS feed.
 * HttpURLConnection is used to fetch RSS feed and kXML is used on xml parsing.
 */
public class RssFeedParser {

    private RssFeed m_rssFeed; // The RSS feed
    private XmlPullParser m_xmlParser = new KXmlParser(); // The Xml parser

    /** Create new instance of RssDocument */
    public RssFeedParser(RssFeed rssFeed) {
        m_rssFeed = rssFeed;
    }

    /** Return RSS feed */
    public RssFeed getRssFeed() {
        return m_rssFeed;
    }

    /** send a GET request to web server */
    public void parseRssFeed()
    throws IOException, Exception {

```

```

HttpConnection hc = null;
DataInputStream dis = null;
String response = "";
try {
    /**
     * Open an HttpConnection with the Web server
     * The default request method is GET
     */
    hc = (HttpConnection) Connector.open( m_rssFeed.getUrl() );
    hc.setRequestMethod(HttpConnection.GET);
    /** Some web servers requires these properties */
    hc.setRequestProperty("User-Agent",
        "Profile/MIDP-1.0 Configuration/CLDC-1.0");
    hc.setRequestProperty("Content-Length", "0");
    hc.setRequestProperty("Connection", "close");

    /**
     * Get a DataInputStream from the HttpConnection
     * and forward it to kXML parser
     */
    parseRssFeedXml( hc.openInputStream() );
} catch(Exception e) {
    throw new Exception("Error while parsing RSS data: "
        + e.toString());
} finally {
    if (hc != null) hc.close();
    if (dis != null) dis.close();
}
}

/**
 * Nasty RSS feed XML parser.
 * Seems to work with both RSS 1.0 and 2.0.
 */
public void parseRssFeedXml(InputStream is)
    throws IOException, XmlPullParserException {
    /** Initialize item collection */
    m_rssFeed.getItems().removeAllElements();

    /** Initialize XML parser and parse RSS XML */
    KXmlParser parser = new KXmlParser();
    parser.setInput( new InputStreamReader(is) );

    /** RSS item properties */
    String title = null;

```

```

String description = null;
String link = null;

/** <?xml...*/
parser.nextTag();

/** Various tags... Wait for the <item> tag */
parser.require(parser.START_TAG, null, null);
while(!"item".equals(parser.getName())){
    /** Check if document doesn't include any item tags */
    if( parser.next() == parser.END_DOCUMENT )
        throw new IOException("No items in RSS feed!");
}

/** Parse <item> tags */
do {
    parser.require(parser.START_TAG, null, null);

    /** Initialize properties */
    title = "";
    description = "";
    link = "";

    /** One <item> tag handling*/
    while (parser.nextTag() != parser.END_TAG) {
        parser.require(parser.START_TAG, null, null);
        String name = parser.getName();
        String text = parser.nextText();

        /** Save item property values */
        if (name.equals("title"))
            title = text;
        else if (name.equals("description"))
            description = text;
        else if (name.equals("link"))
            link = text;

        parser.require(parser.END_TAG, null, name);
    }

    /** Debugging information */
    System.out.println ("Title:    " + title);
    System.out.println ("Link:      " + link);
    System.out.println ("Description: " + description);

    /** Create new RSS item and add it do RSS document's item

```

```

        * collection
        */
        RssItem rssItem = new RssItem(title, link, description);
        m_rssFeed.getItems().addElement( rssItem );
        parser.nextTag();

    } while("item".equals(parser.getName()));
}
}

/**
 * RssItem.java
 */

package RssReader;

/**
 * RssItem class is a data store for a single item in RSS feed.
 * One item consist of title, link and description.
 */
public class RssItem {

    private String m_title = ""; // The RSS item title
    private String m_link = ""; // The RSS item link
    private String m_desc = ""; // The RSS item description

    /** Creates a new instance of RssItem */
    public RssItem(String title, String link, String desc) {
        m_title = title;
        m_link = link;
        m_desc = desc;
    }

    /** Get RSS item title */
    public String getTitle(){
        return m_title;
    }

    /** Get RSS item link address */
    public String getLink(){
        return m_link;
    }

    /** Get RSS item description */
    public String getDescription(){
        return m_desc;
    }
}

```

```

    }
}

/**
 * RssReaderMIDlet.java
 */

package RssReader;

import java.util.*;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

/**
 * RSS feed reader MIDlet
 *
 * RssReaderMIDlet is an application that can read RSS feeds. User can store
 * multiple RSS feeds as bookmarks into application's record store.
 */
public class RssReaderMIDlet extends MIDlet
    implements CommandListener, Runnable {

    // Attributes
    private Display    m_display;    // The display for this MIDlet
    private Settings  m_settings;    // The settings
    private Hashtable  m_rssFeeds;    // The bookmark URLs
    private Thread     m_netThread;   // The thread for networking
    private boolean    m_getPage;     // The noticy flag for HTTP

    // Currently selected bookmark
    private int        m_curBookmark; // The currently selected item
    private RssFeedParser m_curRssParser; // The currently selected RSS

    // GUI items
    private List       m_bookmarkList; // The bookmark list
    private List       m_headerList;  // The header list
    private Form       m_itemForm;     // The item form
    private Form       m_addNewBMForm; // The add new bookmark form
    private Form       m_loadForm;     // The "loading..." form
    private TextField  m_bmName;       // The RSS feed name field
    private TextField  m_bmURL;       // The RSS feed URL field

    // Commands
    private Command    m_addOkCmd;     // The OK command
    private Command    m_addCancelCmd; // The Cancel command
    private Command    m_exitCommand;  // The exit command

```

```

private Command m_addNewBookmark; // The add new bookmark command
private Command m_openBookmark; // The open bookmark command
private Command m_editBookmark; // The edit bookmark command
private Command m_delBookmark; // The delete bookmark command
private Command m_backCommand; // The back to header list command
private Command m_openHeaderCmd; // The open header command
private Command m_backHeaderCmd; // The back to bookmark list command
private Command m_updateCmd; // The update headers command

public RssReaderMIDlet() {
    m_display = Display.getDisplay(this);

    /** Initialize commands */
    m_addOkCmd = new Command("OK", Command.OK, 1);
    m_addCancelCmd = new Command("Cancel", Command.CANCEL, 2);
    m_backCommand = new Command("Back", Command.SCREEN, 1);
    m_exitCommand = new Command("Exit", Command.SCREEN, 4);
    m_addNewBookmark = new Command("Add new", Command.SCREEN, 2);
    m_openBookmark = new Command("Open", Command.SCREEN, 1);
    m_editBookmark = new Command("Edit", Command.SCREEN, 2);
    m_delBookmark = new Command("Delete", Command.SCREEN, 3);
    m_openHeaderCmd = new Command("Open", Command.SCREEN, 1);
    m_backHeaderCmd = new Command("Back", Command.SCREEN, 2);
    m_updateCmd = new Command("Update", Command.SCREEN, 2);

    m_getPage = false;
    m_curBookmark = -1;

    /** Initialize GUI items */

/**
 * Settings.java
 */

package RssReader;

import java.io.*;
import java.util.*;
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;
import javax.microedition.rms.*;

/**
 * A class for storing and retrieving application settings and properties.
 * Class stores all settings into one Hashtable variable. Hashtable is loaded

```

```
* from RecordStore at initialization and it is stored back to the RecordStore
* with save method.
*/
```

```
public class Settings {

    private static Settings m_store;
    private MIDlet      m_midlet;
    private boolean     m_valuesChanged = false;
    private Hashtable   m_properties = new Hashtable();

    /**
     * Singleton pattern is used to return
     * only one instance of record store
     */
    public static synchronized Settings getInstance( MIDlet midlet )
    throws IOException, RecordStoreException {
        if( m_store == null ) {
            m_store = new Settings( midlet );
        }
        return m_store;
    }

    /** Constructor */
    private Settings( MIDlet midlet )
    throws IOException, RecordStoreException {
        m_midlet = midlet;
        load();
    }

    /** Return true if value exists in record store */
    private boolean exists( String name ) {
        return getProperty( name ) != null;
    }

    /** Get property from Hashtable*/
    private synchronized String getProperty( String name ) {
        String value = (String) m_properties.get( name );
        if( value == null && m_midlet != null ) {
            value = m_midlet.getAppProperty( name );
            if( value != null ) {
                m_properties.put( name, value );
            }
        }
        return value;
    }
}
```

```

/** Get boolean property */
public boolean getBooleanProperty( String name, boolean defaultValue ) {
    String value = getProperty( name );
    if( value != null ) {
        return value.equals( "true" ) || value.equals( "1" );
    }
    return defaultValue;
}

```

```

/** Get integer property */
public int getIntProperty( String name, int defaultValue ) {
    String value = getProperty( name );
    if( value != null ) {
        try {
            return Integer.parseInt( value );
        } catch( NumberFormatException e ) {
        }
    }
    return defaultValue;
}

```

```

/** Get string property */
public String getStringProperty( String name, String defaultValue ) {
    Object value = getProperty( name );
    return ( value != null ) ? value.toString() : defaultValue;
}

```

```

/** Load properties from record store */
private synchronized void load()
throws IOException, RecordStoreException {
    RecordStore rs = null;
    ByteArrayInputStream bin = null;
    DataInputStream din = null;

    m_valuesChanged = false;
    m_properties.clear();

    try {
        rs = RecordStore.openRecordStore("Store", true );
        if( rs.getNumRecords() == 0 ) {
            rs.addRecord( null, 0, 0 );
        } else {
            byte[] data = rs.getRecord( 1 );
            if( data != null ) {
                bin = new ByteArrayInputStream( data );
                din = new DataInputStream( bin );
            }
        }
    }
}

```

```

        int num = din.readInt();
        while( num-- > 0 ) {
            String name = din.readUTF();
            String value = din.readUTF();
            m_properties.put( name, value );
        }
    }
} finally {
    if( din != null ) {
        try { din.close(); } catch( Exception e ){}
    }

    if( rs != null ) {
        try { rs.closeRecordStore(); } catch( Exception e ){}
    }
}
}

```

```

/** Save property Hashtable to record store */
public synchronized void save( boolean force )
    throws IOException, RecordStoreException {
    if( !m_valuesChanged && !force ) return;

```

```

    RecordStore rs = null;
    ByteArrayOutputStream bout = new
        ByteArrayOutputStream();
    DataOutputStream dout = new
        DataOutputStream( bout );

```

```

    try {
        dout.writeInt( m_properties.size() );
        Enumeration e = m_properties.keys();
        while( e.hasMoreElements() ) {
            String name = (String) e.nextElement();
            String value = m_properties.get( name ).toString();
            dout.writeUTF( name );
            dout.writeUTF( value );
        }

```

```

        byte[] data = bout.toByteArray();

```

```

        rs = RecordStore.openRecordStore( "Store", false );
        rs.setRecord( 1, data, 0, data.length );
    } finally {
        try { dout.close(); } catch( Exception e ){}

```

```
        if( rs != null ) {
            try { rs.closeRecordStore(); } catch( Exception e ){}
        }
    }
}

/** Set a boolean property */
public void setBooleanProperty( String name, boolean value ) {
    setStringProperty( name, value ? "true" : "false" );
}

/** Set an integer property */
public void setIntProperty( String name, int value ) {
    setStringProperty( name, Integer.toString( value ) );
}

/** Set a string property */
public synchronized boolean setStringProperty( String name, String value ) {
    if( name == null && value == null ) return false;
    m_properties.put( name, value );
    m_valuesChanged = true;
    return true;
}
}
```
