

IMPLEMENTASI PUSH CONTENT WIDGET UNTUK SISTEM INFORMASI JADWAL BIOSKOP PADA PLATFORM ANDROID

Muhammad Aditya Haferush¹, Aries Pratiarso, ST. MT², Achmad Subhan Khalilullah, ST³

¹Politeknik Elektronika Negeri Surabaya, Jurusan Teknik Telekomunikasi

²Laboratorium *Basic Telecommunication*, Politeknik Elektronika Negeri Surabaya
Kampus ITS, Surabaya 60111

e-mail : aditya.haferush@gmail.com e-mail : aries@eepis-its.edu e-mail : subhankh@eepis-its.edu

Abstrak

Menonton film di bioskop merupakan salah satu kebutuhan akan hiburan masyarakat saat ini, namun informasi film terbaru hanya didapat melalui koran terbaru ataupun *website*. Sehingga dibutuhkan sebuah sistem informasi yang mengambil data secara otomatis dari *website* sumber dan kemudian ditampilkan pada ponsel berbasis *Android*.

Kenapa memilih *Android* sebagai *platform* yang digunakan, hal ini disebabkan maraknya penggunaan ponsel dengan sistem operasi tersebut. Dalam proses pembuatan aplikasi ini digunakan pemrograman berbasis *java* khusus untuk *platform Android* dengan didukung oleh *SDK Android* dan *Eclipse* sebagai *development environment*, selain itu *Android* juga mendapatkan dukungan penuh dari pemiliknya, yakni *google* sehingga memudahkan para pengembang aplikasi untuk membuat aplikasi *Android*.

Dari sistem yang telah dibuat dihasilkan suatu aplikasi yang dapat menampilkan informasi film yang akan ditayangkan, lokasi teater, jadwal film serta film yang sedang ditayangkan. Semua informasi diperoleh dari www.blitzmegaplex.com, dari aplikasi yang dijalankan diketahui bahwa kecepatan data dengan *SAX Parser* jauh lebih cepat dibandingkan *htmlcleaner*. Pemilihan sistem *parsing* yang digunakan harus disesuaikan dengan kebutuhan aplikasi yang ada, sehingga didapat keluaran yang optimal.

Kata kunci : *android, java, eclipse, film, google*.

1. Pendahuluan

Pada zaman yang serba modern saat ini, *smartphone* bukanlah suatu perangkat yang hanya dimiliki oleh kalangan eksekutif saja namun sudah sampai pada kalangan menengah. Pada dasarnya sebuah *smartphone* memiliki sebuah sistem operasi untuk menjalankan aplikasi yang ada. Sistem operasi yang saat ini sedang disukai oleh masyarakat adalah *Android*, pada awal kemunculan sistem operasi ini mendapatkan dukungan dari masyarakat luas karena mengusung konsep *open source*. Dengan konsep tersebut maka setiap pengguna dapat mengembangkan aplikasi secara individu.

Penggunaan *smartphone* untuk mengakses layanan internet sangatlah besar, baik hanya untuk membuka akun jejaring sosial, memeriksa *e-mail*, membaca berita hari ini ataupun melihat film terbaru yang akan atau sedang ditayangkan di bioskop kesayangan. *Smartphone* bersistem operasi *Android* atau disebut sebagai ponsel *Android* memiliki banyak kemampuan, mulai dari konektivitas, kenyamanan serta kemudahan dalam mengakses fitur-fitur yang diberikan membuat ponsel ini menjadi primadona bagi kaula muda maupun tua.

Dari salah satu kebutuhan tersebut didapatkan suatu permasalahan baru yakni setiap kali pelanggan membuka halaman dari <http://www.blitzmegaplex.com> maka hal tersebut sangat tidak efisien karena *user* harus membuka aplikasi *web browser* pada ponsel *Android* terlebih dahulu. Untuk itu penulis ingin membuat suatu aplikasi yang bernama *Cinema Information System on Android* yang kemudian disebut dengan "*Cisandro*", aplikasi ini akan membantu pengguna dalam mendapatkan informasi film terbaru yang sedang ditayangkan di bioskop-bioskop yang ada di Indonesia secara otomatis pada ponsel pengguna.

2. Teori Penunjang

2.1 *Android*

Android adalah sistem operasi untuk *smartphone* yang dibuat oleh *google corporation*. Sistem operasi ini dikembangkan dengan memanfaatkan *linux kernel*. Versi *stable* terbaru dari *Android* adalah *Honeycomb 3.0* yang dirilis pada kuartal pertama 2011 dengan *kernel linux* versi 2.6.37 pada versi ini fitur *Android* sudah mendukung *processor multi core*.

Tabel 2.1 Daftar rilis versi *Android*

Nama Kode <i>Android</i>	Kernel <i>Linux</i>	Tahun Rilis
<i>Cupcake 1.5</i>	2.6.27	Februari 2009
<i>Donut 1.6</i>	2. 6.29	September 2009
<i>Éclair 2.0/2.1</i>	2.6.29	Oktober 2009
<i>Froyo 2.2</i>	2.6.32	Mei 2010
<i>Gingerbread</i>	2.6.33/34	Oktober 2010
<i>Honeycomb</i>	2.6.37/38	Q1 2011

Dari arsitektur sistem, *Android* merupakan sekumpulan *framework* dan *virtual machine* yang berjalan di atas *kernel linux*. *Virtual machine Android* bernama *Dalvik Virtual Machine (DVM)*, *engine* ini berfungsi untuk menginterpretasikan dan menghubungkan seluruh kode mesin yang digunakan oleh setiap aplikasi dengan *kernel Linux*. Sementara untuk *framework* aplikasi sebagian besar dikembangkan oleh *google* dan sebagian yang lain dikembangkan oleh pihak ketiga. Beberapa *framework* yang dikembangkan oleh *Android* sendiri misalnya fungsi untuk teleponi seperti panggilan telepon, *sms*, *video call*. Untuk *browser Android* menggunakan *google chrome* yang sebelumnya sudah dikembangkan oleh *google* jauh sebelum *Android* rilis.

Lapisan arsitektur *Android* :

1. *Linux Kernel*

Inti layanan (termasuk *driver* perangkat keras, proses dan manajemen memori, keamanan, jaringan dan manajemen daya) yang ditangani oleh *kernel linux 2.6*. *Kernel* juga menyediakan lapisan abstraksi antara *hardware* dan sisa dari *stack memory*.

2. *Libraries*

Berjalan di atas *kernel*, *Android* mencakup C/C++ dan berbagai *libraries* inti seperti *libc* dan *SSL*, serta :

1. *Media Library* untuk pemutaran media audio dan video.
2. *Surface manager* untuk mendukung *display management*.
3. *Graphics libraries* yang menyertakan *SGL* dan *OpenGL* untuk 2D dan 3D.
4. *SQLite* untuk dukungan *database*.
5. *SSL* dan *WebKit* untuk *web browser* dan keamanan *internet* yang terintegrasi.

3. *Android Run Time*

Terdiri dari *Core libraries* dan *Dalvik virtual machine*, *Android Run Time* merupakan sebuah mesin yang mendukung aplikasi dan bersama dengan *libraries* membentuk dasar dan kerangka dari aplikasi tersebut.

4. *Application Framework*

Bagian ini menyediakan *class* yang digunakan untuk membuat aplikasi *Android*. Disediakan pula *generic abstraction* untuk akses perangkat keras dan mengatur *user interface* serta sumber daya dari suatu aplikasi.

5. *Application Layer*

Semua aplikasi baik dari pihak ketiga ataupun milik *google* sama-sama dibangun menggunakan *API Libraries* yang sama. Lapisan dari tiap aplikasi berjalan pada *Android run time* dengan menggunakan *class* dan *services* yang disediakan oleh *application framework*. *Android* memiliki empat komponen meliputi *activity*, *broadcast receiver*, *services*, dan *content provider*.

Komponen aplikasi dapat disebut juga sebagai elemen-elemen aplikasi yang bisa dikembangkan pada *platform Android*

2.2 Parsing System

Parsing adalah suatu proses menganalisa suatu kumpulan kata dengan memisahkan kata tersebut dan menentukan struktur sintaksis dari tiap kata tersebut. Gramatika yang dipakai sangat berkaitan dengan proses parsing apa yang digunakan. Pada *Bottom-Up parsing* gramatika yang dipakai akan lebih banyak bercabang ke arah simbol *non-terminal*. Hal lain yang juga berkaitan dengan proses parsing adalah kamus atau *leksikon* yang digunakan. Dalam leksikon disimpan daftar kata yang dapat dikenali sebagai simbol terminal dalam grammar dan informasi yang diperlukan untuk tiap kata tersebut dalam proses parsing yang bersangkutan.

Dari pendekatan dalam mengenali struktur suatu kalimat, proses parsing dapat dibagi menjadi dua bagian besar yaitu *Top Down parsing* dan *Bottom Up Parsing*. *Top Down Parser* memulai pemeriksaan dari simbol awal *s* dan mencoba untuk mencari bentuk simbol terminal berikutnya yang sesuai dengan jenis kata dari kalimat masukan. Cara sebaliknya diterapkan *Bottom Up parser* yaitu mencari dari simbol-simbol terminal menuju ke arah pembentukan simbol awal.

2.2.1 SAX Parser

SAX (Simple Api for XML) berbasis pada urutan *event* yang diakses oleh *API parser*, dikembangkan oleh XML-DEV. *SAX* menyediakan mekanisme untuk data dari dokumen XML yang menjadi alternatif dari *Document Object Model (DOM)*. Dimana *DOM* beroperasi pada dokumen secara keseluruhan, sedangkan *SAX* beroperasi pada setiap bagian dari dokumen XML secara berurutan.

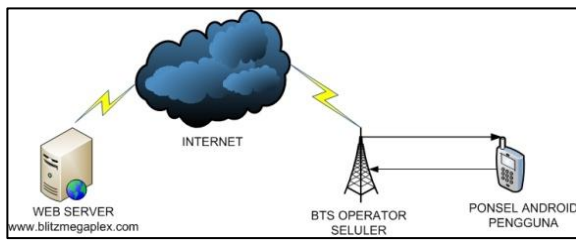
2.2.2 Htmlcleaner

HtmlCleaner adalah *open-source* HTML parser yang ditulis di *Java*. HTML pada *web* biasanya tidak sesuai dan memiliki format yang tidak baik serta tidak cocok untuk pemrosesan lebih lanjut. Untuk kebutuhan yang penting seperti dokumen, perlu dilakukan perbaikan terhadap kerancuan kode serta mengurutkannya berdasarkan *tag* sesuai atribut teks biasa. Pada dokumen HTML yang diberikan, maka *HtmlCleaner* akan melakukan penyusunan ulang pada tiap elemen dan mengubahnya menjadi format XML yang tersusun rapi. Pada kondisi normal, hal ini mengikuti aturan yang biasanya digunakan pada sebagian besar *web browser* dalam membuat *Document Object Model*. Namun, pengguna dapat memodifikasi *tag* yang ada agar memiliki keluaran yang sesuai kehendak.

3. Perancangan Sistem

Untuk menyelesaikan pembuatan sistem ini maka dilakukan perancangan terhadap sistem yang akan

dibuat. Agar sistem dapat bekerja dengan baik dibutuhkan perencanaan sebagai berikut :



Gambar 3.1. Blok Diagram sistem keseluruhan

Pada sistem ini terdapat tiga blok bagian dari keseluruhan sistem, pertama adalah blok pada sisi pengguna aplikasi. Aplikasi yang akan dibangun berjalan pada sistem operasi *Android 2.2 (Froyo)*. Blok kedua merupakan sisi *provider* seluler, bagian ini bertugas menghubungkan aplikasi pengguna dengan *web server* www.blitzmegaplex.com via koneksi *internet*. Blok ketiga merupakan *web server* www.blitzmegaplex.com yang berperan sebagai sumber data yang akan dikirimkan ke pengguna.

Proses berjalannya sistem ini dimulai dengan disajikannya beberapa pilihan menu pada *CISANDRO*, yakni *Location*, *Now Playing*, *Coming Soon* dan *About*. Menu *Location* akan mengirimkan *http-request* ke <http://www.blitzmegaplex.com/rss/schedule.php>.

Alamat tersebut merepresentasikan nama bioskop milik *Blitzmegaplex* beserta alamatnya. Selanjutnya permintaan tadi dilewatkan ke koneksi *internet* via operator seluler yang digunakan pengguna, untuk kemudian diteruskan ke *web server* www.blitzmegaplex.com dan kemudian ditujukan ke *file schedule.php* yang merupakan *file RSS Feed* dari *blitzmegaplex*. *File* ini kemudian menjadi input dari *SAX Parser* yang sudah terintegrasi dengan *Android*, dan sebagai outputnya adalah daftar bioskop milik *blitzmegaplex* yang ada di Jakarta dan sekitarnya serta film yang tayang pada bioskop tersebut.

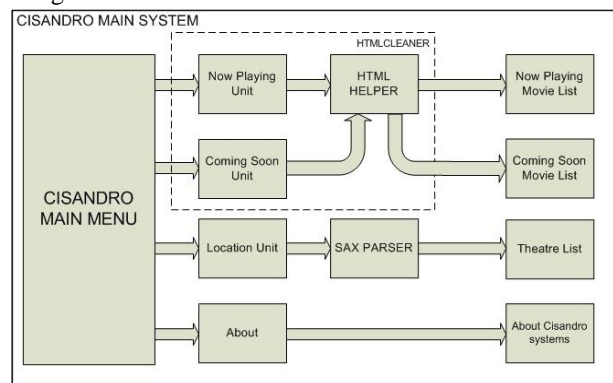
Pada menu *location* target halamannya adalah http://www.blitzmegaplex.com/en/schedule_today.php. Saat *web server* merespon permintaan tadi maka yang diterima pada sisi pengguna berupa *source code* dari *schedule_today.php*. Kemudian dilakukan *parsing* terhadap file tersebut hingga pada akhirnya tersisa judul-judul film yang diputar pada hari itu. Proses ini juga terjadi pada menu *coming soon*, perbedaannya hanya pada *file* yang diakses yakni *schedule_comingsoon.php*.

Menu terakhir adalah *About*, pada bagian ini pengguna akan diberikan penjelasan singkat mengenai aplikasi *cisandro*, tentang cara kerja, *developer* dan hal-hal yang diperlukan untuk informasi bagi pengguna.

4. Pembuatan dan Hasil

Dari rancangan sistem yang dibuat maka dilakukan pembangunan sistem dengan motode subsistem yang nantinya diintegrasikan menjadi satu

buah sistem utama sebagaimana dapat dilihat pada blok diagram dibawah ini.



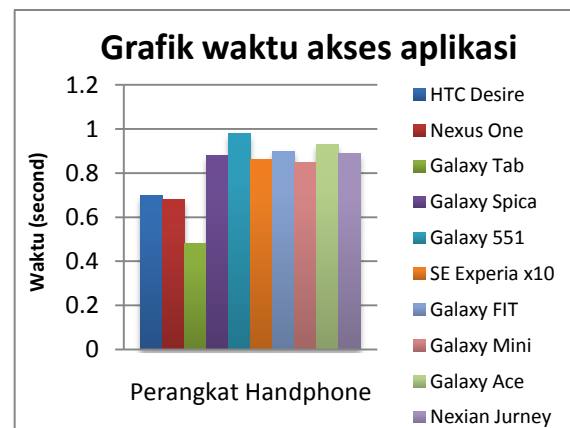
Gambar 4.1. Blok diagram sistem yang dibangun

Dari sistem di atas dapat dilakukan 2 pengujian, yakni keberhasilan pengguna dalam menjalankan aplikasi, serta kecepatan tiap menu dalam menampilkan informasi yang ada dengan parameter masukan berupa *file html* dan *xml*.

Pengujian pertama dilakukan dengan menggunakan 10 perangkat ponsel *Android*. Pengujian dilakukan dengan menjalankan aplikasi pada tiap ponsel *Android*, hasil yang didapat dari pengujian ini adalah seluruh perangkat berhasil menjalankan aplikasi yang dibuat dengan waktu yang tidak sampai 1 detik.

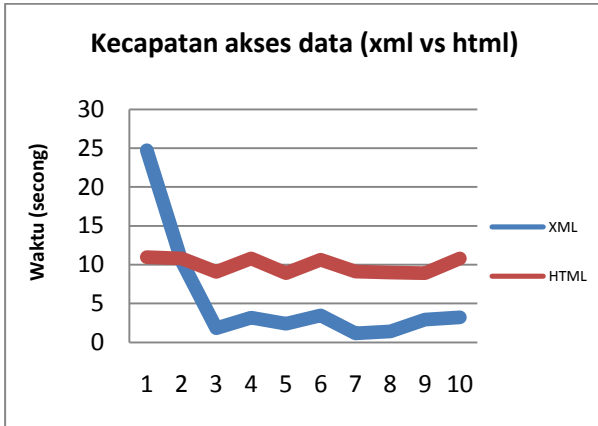
Tabel 4.1 Pengujian aplikasi pada ponsel *Android*

Ponsel	Waktu (second)	Status	
		Berhasil	Gagal
HTC Desire	0.70	√	
Nexus One	0.68	√	
Galaxy Tab	0.48	√	
Galaxy Spica	0.88	√	
Galaxy 551	0.98	√	
SE Xperia x10	0.86	√	
Galaxy Fit	0.90	√	
Galaxy Mini	0.85	√	
Galaxy Ace	0.93	√	
Nexian Journey	0.89	√	



Gambar 4.2. Grafik waktu akses aplikasi

Pengujian selanjutnya adalah pengukuran kecepatan dalam menampilkan informasi dimulai dari ditekannya tombol menu sampai dengan tampilnya informasi yang diinginkan, pengujian ini dilakukan sebanyak 10 kali untuk masukan dengan *file xml* dan *html*. Pada Gambar 4.3 terlihat bahwa masukan dengan *file xml* lebih cepat dibandingkan *html*.



Gambar 4.3. Grafik kecepatan akses data

5. Kesimpulan

Dari hasil pengamatan selama perancangan, pembuatan hingga pengujian sistem dapat disimpulkan bahwa waktu yang diperlukan untuk menjalankan aplikasi *cisandro* tidak lebih dari 1 detik. Pada sisi kecepatan data (dalam waktu) masukan dengan *file xml* lebih cepat dibandingkan dengan masukan *file html*, hal ini disebabkan perbedaan *parsing system* yang digunakan. Sehingga dapat dinyatakan bahwa *SAX Parser* lebih cepat dibandingkan dengan *htmlcleaner*.

Daftar Pustaka

- [1] Meier Reto, “*Professional Android Application Development*”, Wiley Publishing. Inc, Indianapolis, 2009.
- [2] Murphy, Mark L, “*Begining Android*”, Apress, 2010.
- [3] Hashimi, Sayed Y. and Satya Komatineni, “*Pro Android*”, Springer-Verlag, New York, 2009.
- [4] Refly Puspitasari Bayudini, “*Rancang Bangun Sistem Informasi Bioskop Online*”, Proyek Akhir Pens-ITS, 2009.