

Membangun *Wireless Application* Menggunakan Teknologi J2ME

Faisal Wiryasantika

faisal@winwinfaisal.info

<http://faisal.winwinfaisal.info>

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di **IlmuKomputer.Com** dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari **IlmuKomputer.Com**.

Teknologi Java merupakan sebuah teknologi yang berkembang sangat pesat akhir-akhir ini. Bahkan belakangan ini dikabarkan berusaha mengalahkan Microsoft yang terkenal sebagai kampiun dari produsen *operating system* dimuka bumi ini. Teknologi Java yang pada awalnya dikenal untuk aplikasi pada *desktop* (J2SE) ataupun pada *application server* (J2EE), kini hadir dengan teknologi terbarunya, J2ME™ Platform, untuk pembangunan aplikasi pada *mobile device* seperti *mobile phone* dan PDA. Selain J2ME™ Platform yang termasuk baru, terdapat pula satu buah *platform* teknologi Java yang termasuk baru pula yaitu Java Card. Java Card merupakan seperangkat kakas untuk membangun aplikasi pada sebuah *card electronic* seperti SIM Card pada ponsel kita. SIM Card yang dipergunakan oleh *Mobile Banking* BCA (layanan operator selular Excelcomindo) menjadi salah satu contoh dari teknologi Java Card.

Tulisan ini ditujukan untuk memberikan *overview* tentang J2ME™ Platform, dan diskursus J2ME Configurations & Profiles yang digunakan untuk membangun aplikasi Java untuk *mobile phones*, PDA, dan *two-way pagers*.

The Network is the Computer™

Computing atau komputasi ada dimana-mana! Kita masih dapat menemukan *one-computer-to-many-people* (mainframes) dan *one-person-to-one-computer* (PCs) di berbagai institusi atau organisasi diberbagai belahan dunia. Belakangan terdapat aplikasi yang sangat menarik yaitu *mobile application* yang dapat hadir dimanapun kita berada, ketika belanja, ketika menyetir sekalipun, bahkan ketika duduk disudut dapur anda, aplikasi ini dapat hadir dan siap membantu anda seolah dunia telah berada dalam genggam tangan anda ☺.

Apakah *devices* yang menarik bagi para *programmers* dan *engineers* saat ini? Banyak orang berkata bahwa *devices* yang paling diminati adalah perpaduan antara *computing* + *networking*. Sebuah *mobile phone's* memiliki kapabilitas *wireless networking* sekaligus kapabilitas *computing* (walau masih terbatas). Bila anda selama ini hanya dapat *browsing* dihadapan *PC desktop* yang bersifat *wireline*, maka saat ini melalui *mobile devices* anda dapat melakukan *browsing* dengan kemampuan yang tidak mengecewakan dibandingkan *PC desktop*, anywhere!

Sun Microsystems telah melansir bahwa kedepan teknologi yang memadukan antara *networking* dan *computing* akan memberikan potensi yang besar bagi perkembangan teknologi di dunia ini. Perkembangan selanjutnya adalah XML yang telah menjadi standar bagi portable application pada lingkungan Java, dan IP sebagai standar bagi interoperabilitas & skalabilitas networking, maka Sun seolah mengeluarkan mantra “The Network is the Computer™”.

- Open processing: Java™ 2 Platform, Micro Edition (J2ME)
- Open data: XHTML and related XML markups
- Open protocols: IP/TCP/HTTP

Java™ + XML + IP protocols

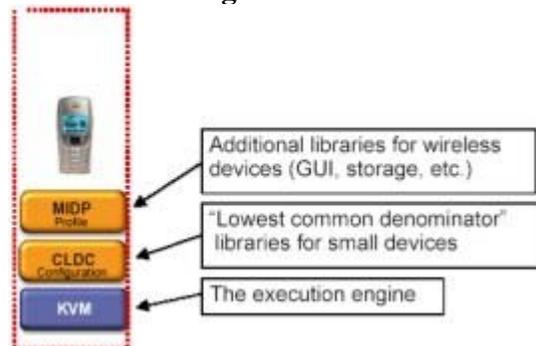
Teknologi Java, XML, dan IP telah membuat Java ada dimana-mana, *multi device computing*. Bahasa Java telah memberikan cara yang konsisten bagi *programmer* agar dapat menulis kode yang dapat dieksekusi pada berbagai *devices*. Untuk *client-side programming*, Visual Basic masih mendominasi pasar dunia, sedangkan untuk pasar *application server* hampir seluruh pasar telah diambil oleh Java, dan kini pun teknologi Java telah diadopsi oleh berbagai vendor *mobile devices* seperti ponsel maupun PDA. Saat tulisan ini dibuat harga sebuah HP Siemens M50 hanyalah seharga 1 juta rupiah, anda telah mendapatkan HP yang *Java enabled* dan siap untuk menemani anda untuk “bercanda” dgn teknologi J2ME.

J2ME and The Java 2 Platform



Anda jangan berharap dapat memprogram dengan *platform* J2SE dan dapat me-load-nya pada *mobile devices*, tentu saja berbeda terutama pada hal *prosesor computing* dan *memory*-nya. Jika pada J2SE anda menggunakan interpreter JVM, pada platform J2ME anda mempergunakan Kilo Virtual Machine (KVM) yang “ditanam” dalam handheld devices. Sebagai informasi bahwa J2EE merupakan *superset* dari J2SE, sebaliknya J2SE merupakan *superset* bagi J2ME. Apa artinya? API yang ada pada J2ME sebagian mengadopsi yang ada pada API J2SE, selain juga mengimplementasikan API spesifik untuk teknologi J2ME itu (`javax.microedition.*`).

What are Configurations and Profiles?



Configuration mendefinisikan minimum Java *Libraries* dan kapabilitas yang dipunyai oleh para developer J2ME. Artinya antara *mobile device* yang *Java enabled* maka akan ditemui configuration yang sama. Bila dianalogikan dengan sekelompok mobil maka sebuah mobil dengan mobil lainnya memiliki kesamaan, misalnya mempunyai roda, kaca spion, speedometer, dll. Roda dan kaca spion itulah yang disebut sebagai configuration. Jadi configuration hanyalah mengatur hal-hal yang berkaitan dengan “kesamaan” bukan mengatur hal-hal yang “membedakan”, sehingga configuration memastikan portabilitas antar *devices*.

Configuration ini ditentukan perkembangannya oleh JCP (Java Community Process), inilah badan non-profit yang berkuat dengan perkembangan teknologi Java. Saat ini telah didefinisikan dua buah configuration yaitu CDC & CLDC dengan perbandingannya sebagai berikut :

Tabel 1. Tabel perbandingan CLDC dan CDC

CLDC (Connected Limited Device Configuration)	CDC (Connected Device Configuration)
Mengimplementasikan subset dari J2SE.	Mengimplementasikan seluruh fitur dari J2SE.
JVM yang digunakan adalah KVM.	JVM yang digunakan adalah CVM.
Digunakan pada perangkat handheld (handphone, PDA, two way pager) dengan memory terbatas (160-512 kb).	Digunakan pada perangkat handheld (internet TV, Nokia Communicator, car TV) dengan memory minimal 2 Mb.
Prosesor : 16/ 32 bit.	Prosesor : 32 bit.

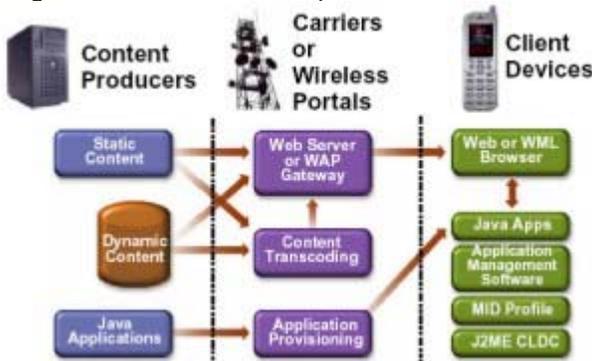
Profile merupakan kebalikan dari configuration yaitu mengatur hal-hal yang spesifik untuk sebuah *device* atau tipe market. Kembali pada analogi mobil diatas, sebuah mobil BMW tentu memiliki ciri spesifik yang tidak dimiliki oleh mobil Mercedes, dan sebaliknya. Misalkan pada profile ini diatur tentang *persistent storage* dan UI. Saat ini JCP telah mendefinisikan lima buah profile, salah satunya yaitu MIDP yaitu profile yang digunakan pada banyak *mobile devices* seperti Siemens M50 dan SL45i.

Web Content for Mobile Devices

Terdapat tiga opsi untuk *me-retrieve static (mark up) content* untuk sebuah *mobile phone* saat ini :

- ❑ cHTML (compact HTML) yaitu merupakan subset dari HTML yang digunakan oleh NTT Docomo sebuah operator selular maha sukses di Jepang hingga teknologinya (I-mode) diekspor ke berbagai negara.
- ❑ WML (wireless markup language) yaitu sebuah markup language untuk lingkungan *wireless* yang didefinisikan oleh WAP Forum. WML digunakan sebagai standar pada *mark-up language* diberbagai negara, meskipun untuk negara-negara Amerika Utara masih didominasi oleh HDML (handheld device markup language).
- ❑ HTML. Jadi terdapat pula *mobile devices* yang dapat menginterpretasi HTML seperti Palm OS yang mempunyai HTML browser.

Bagaimana sebuah content dapat di-*retrieve* oleh *mobile device*?



Pada bagian atas kiri terlihat *content* berisi *static file* atau bisa juga berupa *dynamic content* atau juga *Java Application* yang di-*generated* oleh *server*. Lebih luas lagi, informasi dapat diekstrak dari basis data dan dimodifikasi menggunakan *servlets*, *EJB*, atau teknologi *server side* lainnya.

Selanjutnya kita lihat pada bagian tengah, *Java Application* ditampilkan oleh *Web Server* dan seterusnya ditransmisikan melalui jaringan *wireless* menuju *mobile devices*. *Web server* atau *WAP gateway system* akan mendistribusikan *content* pada *mobile phone* atau *device* lainnya.

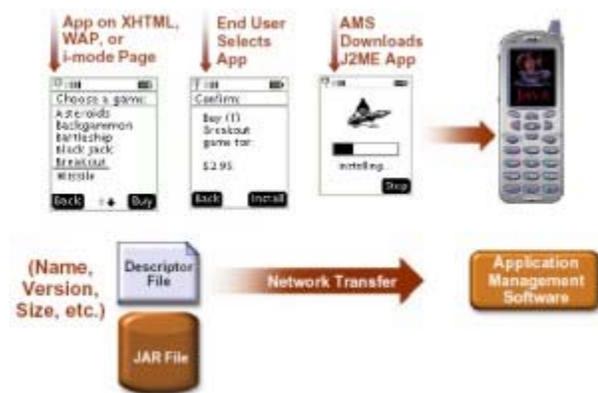
Akhirnya *content* diterima oleh *client device* dengan melakukan proses *transcoding*, baik itu mengambil data dari basis data, atau XML, atau HTML, dan selanjutnya diterjemahkan kedalam format WML atau cHTML bergantung pada format yang kita punyai.

Bagaimana *Java Application* dapat ditransmisikan melalui *jaringan wireless* dan selanjutnya diterima oleh *client device*? Bagaimana pula operator atau *content provider* mengatur masalah *billing* untuk *Java*

Application atau *service* yang telah diberikan? Beberapa *company* telah bekerja untuk mendefinisikan masalah ini dan yang pertama melakukan adalah LG Telecom di Korea pada tahun 2000, disusu NTT DoCoMo di Jepang dan Nextel di USA pada akhir 2001. Setiap operator atau network memiliki ciri spesifik walaupun datang dengan teknologi yang sama yaitu 3G *wireless technology* yang dimulai tahun 2001-2002. Untuk menjembatani perbedaan tersebut maka sebuah MIDP Groups membuat sebuah ketentuan/ standar OTA (*over the air*) yaitu JSR118, walaupun ketentuan OTA ini masih terus disempurnakan.

Java Application Loading Process

Ketika sebuah *application* siap untuk didistribusikan maka dibutuhkan suatu mekanisme agar secara fisik aplikasi itu dapat di-load kedalam *phone*. AMS (*application management software*) adalah kakas pada ponsel yang mengatur mekanisme ini. Jadi AMS ini dapat mengetahui bagaimana cara me-load kode MIDlet melalui *wireless connection*. Berikut adalah proses *loading* sebuah Java *Application* :



Pada skenario disamping, seorang *consumer* akan mendatangi WAP page. Pada halaman tersebut terdapat aplikasi anda yang dapat di-download melalui *mobile set*. Pada saat sebuah aplikasi dipilih maka file jad (java descriptor) akan di-download ke mobile set anda. File jad ini biasanya hanya sekitar ratusan *bytes* sehingga sangat cepat dan murah untuk di-download.

File jad ini akan memberitahukan pada *consumer* tentang hal-hal *basic* tentang aplikasi tersebut (seperti versi maupun *size* dari aplikasi tersebut), sehingga manakala *consumer* telah memiliki versi yang sama ataupun *size*-nya terlalu besar untuk sisa *space* yang anda punyai maka akan memberikan *alert* tertentu.

Sebagai contoh anda ingin men-download sebuah games MIDlet dari sebuah situs, ketika anda men-download jad file akan terlihat ukuran file games tersebut (jar=java archive file) misalkan sebesar 6kb padahal sisa *space memory* anda hanya 2 kb maka akan muncul alert “*user doesn't have enough room to download and store this application*”. Jelas ini menguntungkan *consumer* karena akan menghemat *bandwidth* maupun uang, paling tidak *consumer* tidak membuang waktu untuk sebuah aplikasi yang tak perlu di-download-nya.

Supporting WAP in J2ME Enabled Mobile Phones

Bagaimana J2ME dan WAP berelasi?

WML bekerja baik untuk beberapa aplikasi seperti *text-centric application* seperti *weather report*, *stock quotes*, dll, tetapi terdapat beberapa kelemahan dengan WAP yaitu WAP-based *application* memerlukan koneksi terhadap network secara permanen, artinya koneksi terhadap internet harus terus dilakukan bilamana ingin menjalankan WAP-based *application*. Selain itu pula aplikasi WAP tidak baik digunakan pada aplikasi yang *graphic intensive* seperti games, *interactive charting*, dll. Selain dari itu isu *security* juga menimpa WAP gateway, dimana terjadi konversi protocol yaitu dari protocol internet menjadi protocol *wireless*.

Teknologi Java dapat dijalankan pada *devices* dengan cara *disconnected access* dan inilah yang tidak dimiliki oleh WAP-based *application*. J2ME memberikan kemudahan bagi *developer* untuk membuat aplikasi grafis seperti *color games*, *office application*, dll. J2ME pula menyediakan *end-to-end security* HTTP seperti SSL

Tetapi tidak berarti teknologi Java dan WAP berdiri sendiri dan justru sebaliknya merupakan teknologi komplementer dimana WAP browser (microbrowser) digunakan untuk *surfing* mencari Java *Application* kemudian AMS akan melakukan manajemen instalasi lalu KVM akan menjalankan aplikasi MIDlet secara *disconnected access*.

Saat ini telah tersedia beberapa microbrowser yang ditulis dengan menggunakan bahasa Java. Tentu saja ini menguntungkan karena anda dapat melakukan proses *updating* secara OTA, sedangkan untuk microbrowser yang ditulis dengan bahasa selain Java tidak memiliki kemampuan untuk di-*updated*, dengan kata lain jika anda menginginkan microbrowser versi terbaru anda harus membeli HP baru ☹.

Openwave (merger dari www.software.com dan www.phone.com) yang merupakan *founder* dengan WAP Forum dan merupakan supporter utama dunia WAP telah mengumumkan *major partnership* dengan Sun Microsystems untuk mengintegrasikan J2ME dan WAP untuk *release* selanjutnya.

What's Next?

J2ME CLDC dan KVM telah digunakan oleh banyak *platform* dan *devices*, termasuk oleh Motorola phones dan *two-way pagers*, Research in Motion (RIM) *wireless handhelds*, dan Palm PDA. Jutaan J2ME CLDS digunakan sehari-hari oleh berbagai *mobile phone* di Jepang dan Korea. Banyak pula consumer yang menggunakan J2ME dengan Motorola *handsets* pada jaringan Nextel di USA, dan dalam tahap *test* dan *commercial deployment* di Vodaphone Airtouch, Sprint PCS, Telefonica, dan operator lainnya di USA. Symbian terus *support* Java Technology untuk di-*release*-nya Nokia Communicator 9210 yang berbasiskan Java Technology pada akhir 2001.

Saat ini terdapat berbagai J2ME *developmet tool* yang *free*. Sun menyediakan J2ME Wireless Toolkit (WTK) yang berisi full J2ME CLDC dan MIDP implementation. *Tool kit* ini hanya dapat bertindak sebagai emulator saja (bukan IDE) dan dilengkapi dengan beberapa *generic phone*. WTK ini dapat dijalankan secara *stand alone* maupun terintegrasi dengan IDE buatan Sun Microsystems pula, Forte Community Edition (CE). Selain Forte CE terdapat pula Forte Mobile Edition (ME) yang berdiri sendiri sebagai IDE untuk J2ME *application*. Borland sendiri mengeluarkan pula IDE-nya untuk pembuatan J2ME *application* yaitu Borland Mobile Set. Baik WTK, Forte CE/ ME, Borland Mobile Set adalah kakas yang *free* untuk di download pada masing-masing situs pengembangnya.

It is a Great Time to Become a J2ME Wireless Developer! Get Started Today!



